Text Recognition and 2D/3D Object Tracking

PhD. Defense
Rodrigo Minetto
Advisors: Profs. Dr. Matthieu Cord and Dr. Jorge Stolfi
Co-advisors: Profs. Dr. Nicolas Thome and Dr. Neucimar J. Leite

Institute of Computing (IC) – University of Campinas (UNICAMP)

Laboratoire d'Informatique de Paris 6 (LIP6) – Université Pierre et Marie Curie (UPMC)

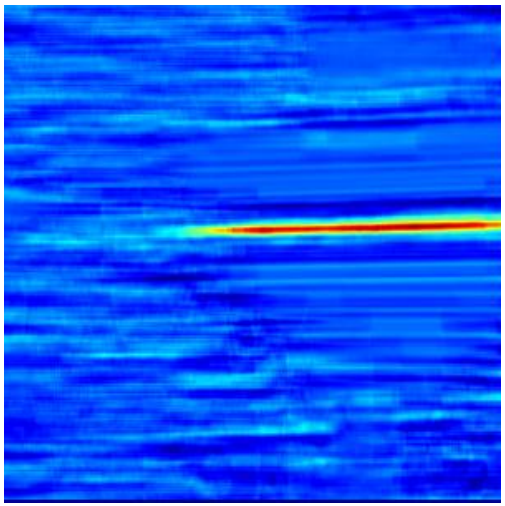
19 March 2012

Overview

1. **Text detection and recognition;**
2. **Text tracking;**
3. **Tracking of 3D rigid objects;**
4. **General conclusions;**
5. **Publications.**

Part I

Text detection and recognition



Contributions

- Description of a novel text classifier (T-HOG):
  - Region splitting into horizontal cells;
  - Cells weighting by fuzzy overlapping functions.
- Comparison of the standard HOG (R-HOG) and T-HOG for text recognition;
- Text filtering and detection with T-HOG;
- Development of a full text detection system:
  - SnooperText;
  - SnooperText + T-HOG: on a real GIS.

Contributions

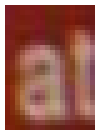
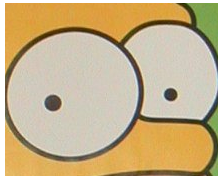
;

Contributions

Statement of the problem

Text recognition (Text/Non-text classification problem)

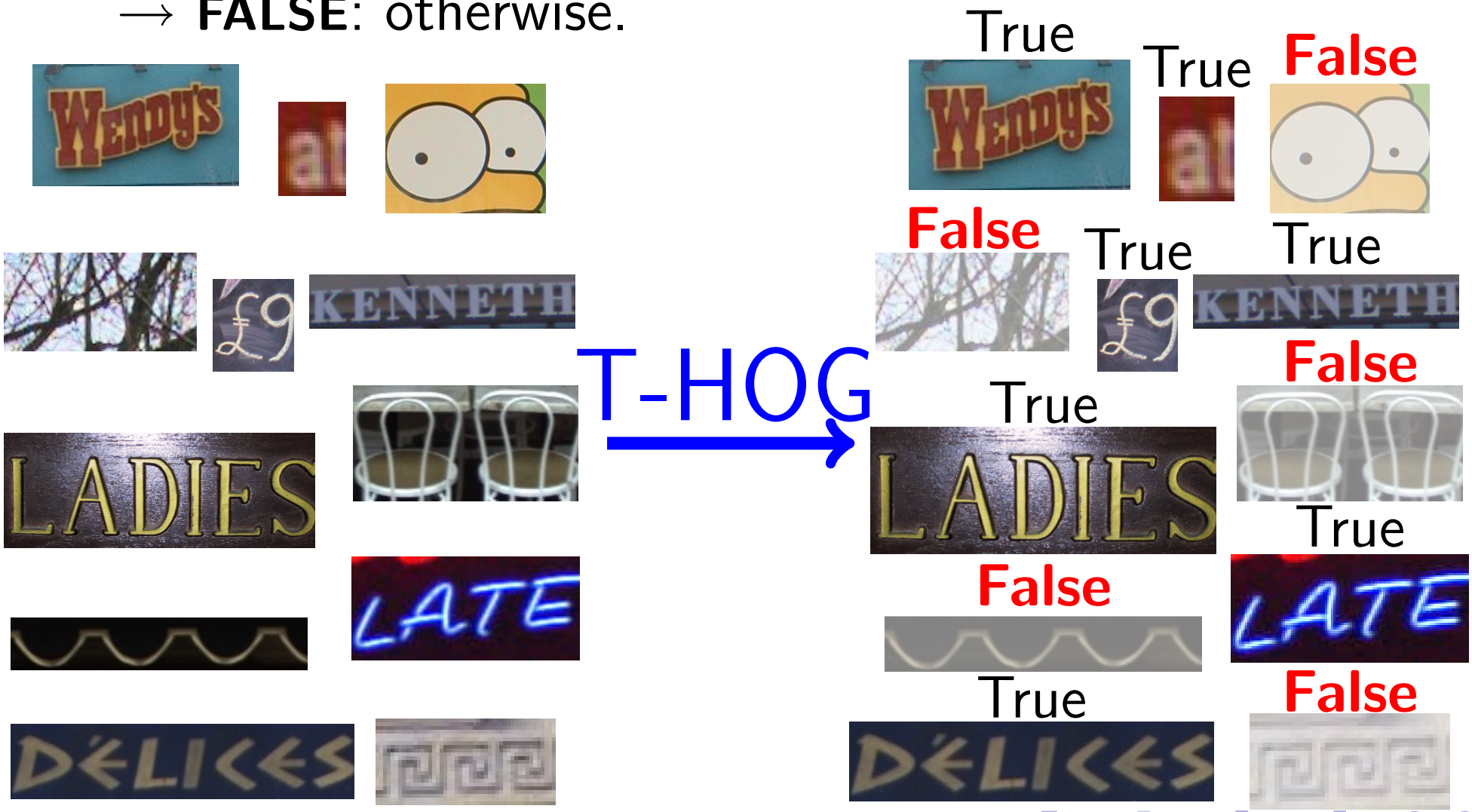
- Input data: sub-images of an arbitrary 3D scene.



Statement of the problem

- Output data: binaries decisions
  - → **TRUE**: if the sub-image contains a Roman-like text;
  - → **FALSE**: otherwise.



R-HOG Idea (Histogram of Oriented Gradients)

- Images of complex objects typically have $\neq$ HOG's in $\neq$ parts;
- Humans: $\neq$ gradient orientation distributions (head/torso/legs):

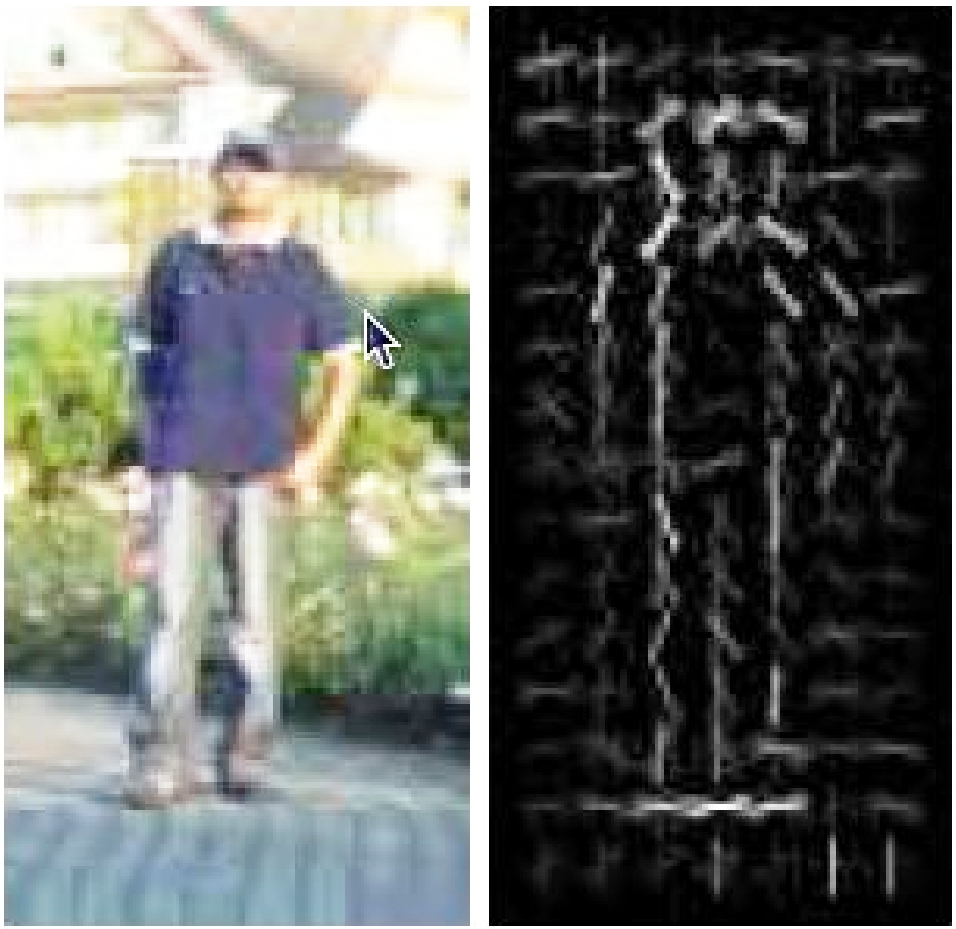
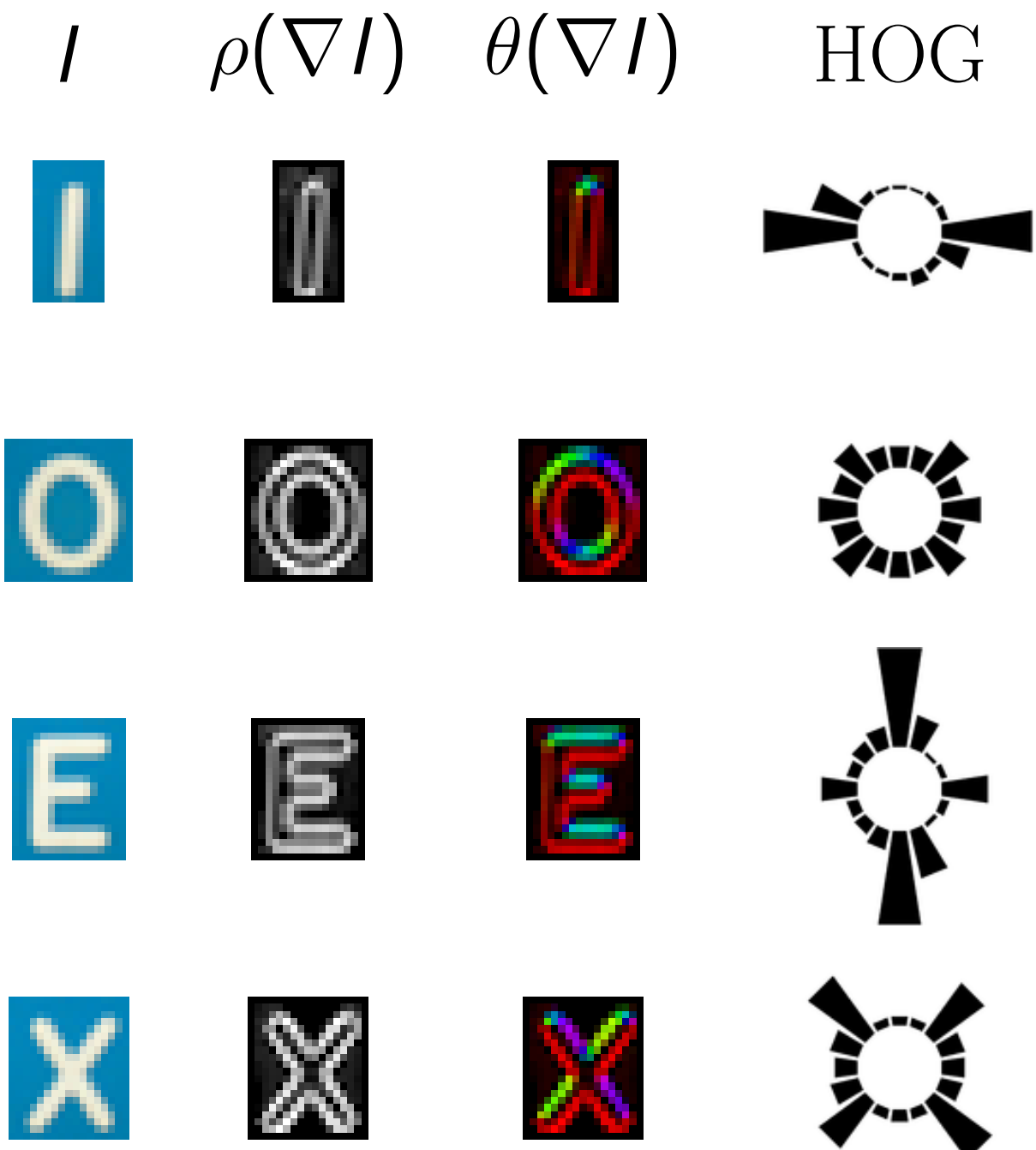


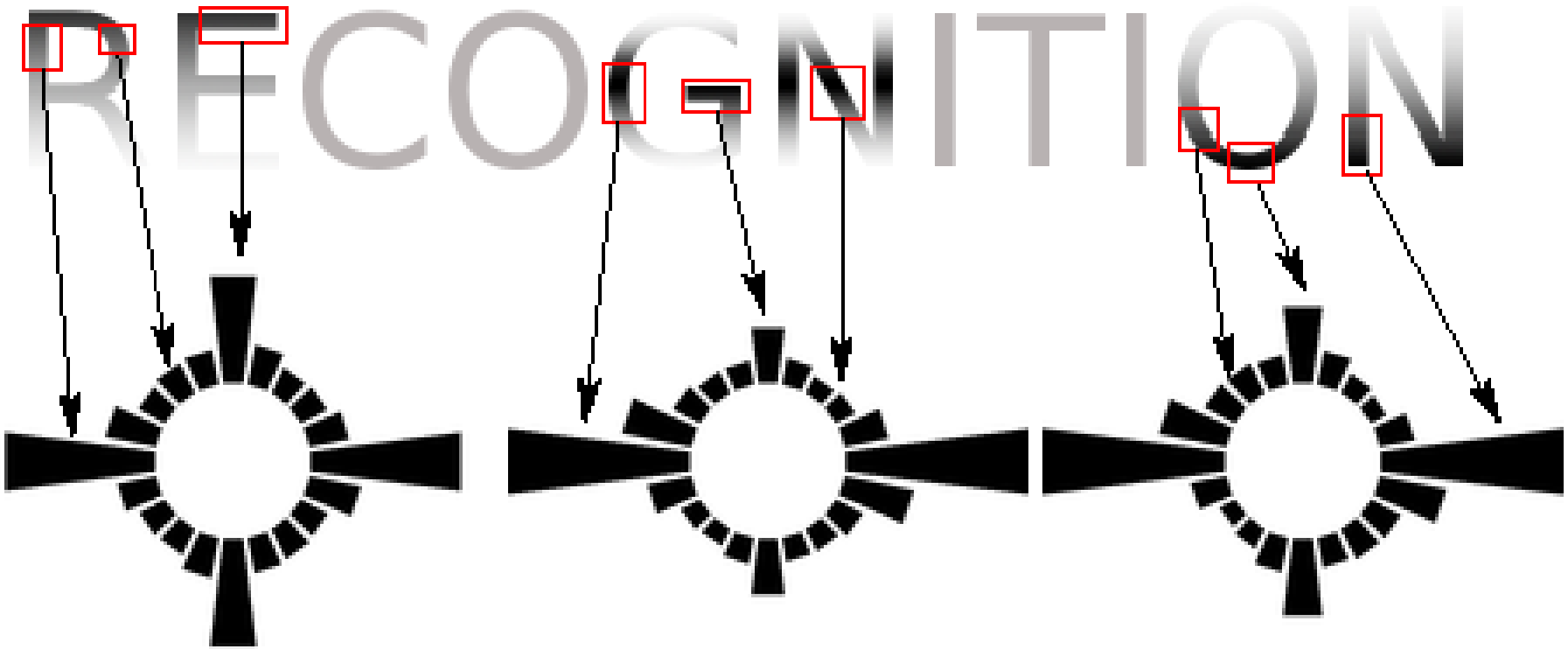
Figure: Image from: Histograms of Oriented Gradients for Human Detection. Navneet Dalal and Bill Triggs. CVPR 2004

HOGs of some isolated letters:

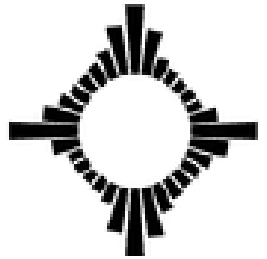


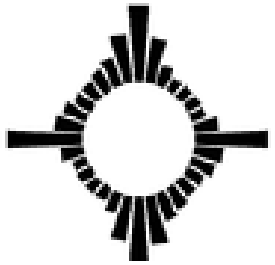
T-HOG Idea

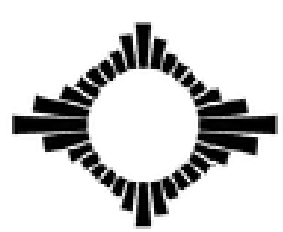
- Roman-like text-lines: $\neq$ HOG's in the top/middle/bottom parts.
- Top/Bottom: **Large proportion of horizontal strokes** $\rightarrow$ gradients pointing mostly in the **vertical** direction;
- Middle: **Large proportion of vertical strokes** $\rightarrow$ gradients pointing mostly in the **horizontal** direction;
- All parts: **Small amount of diagonal strokes**

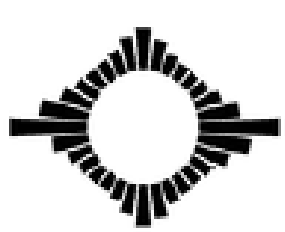


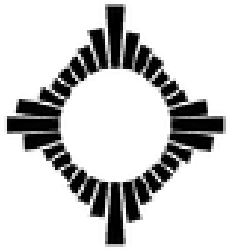
F-HOG of text/non-text regions

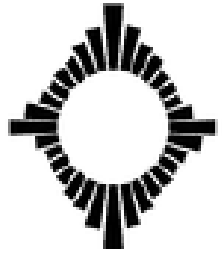




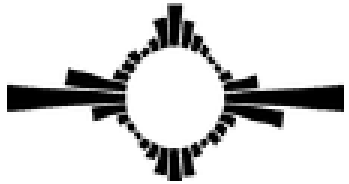


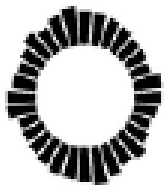


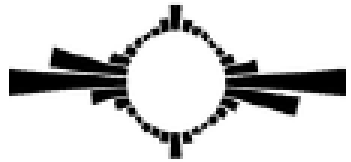


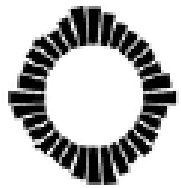


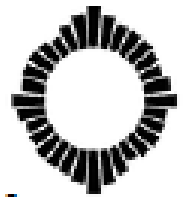
top *middle* *bottom*







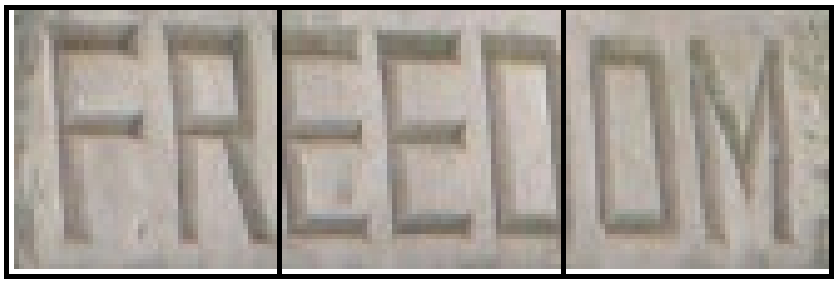




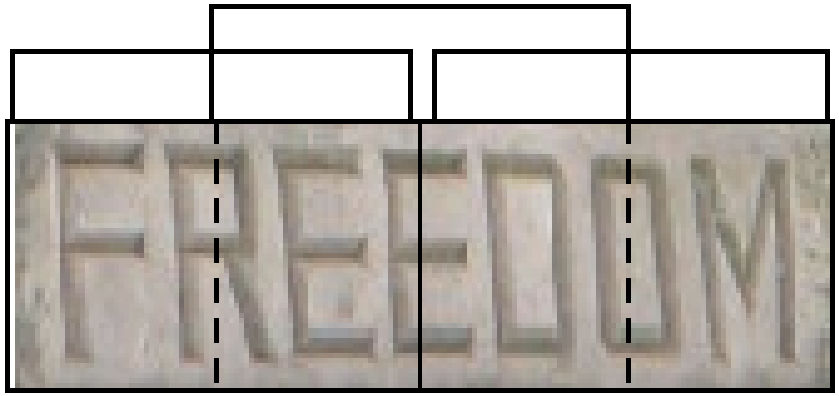
top *middle* *bottom*

Standard HOG blocks/cells Arrangements

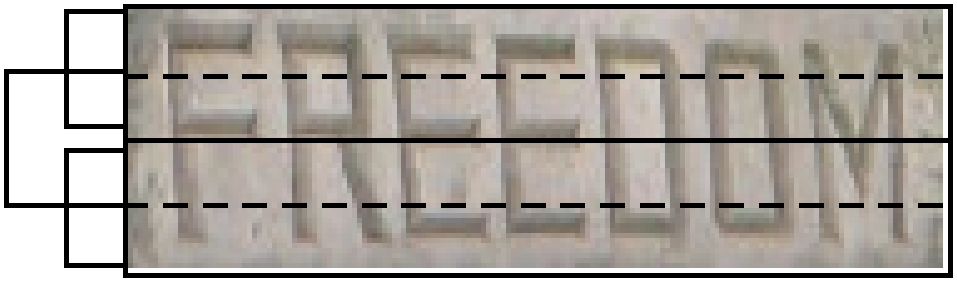
- What is the best region splitting for text recognition?
- Region splitting into 3 HOG's:



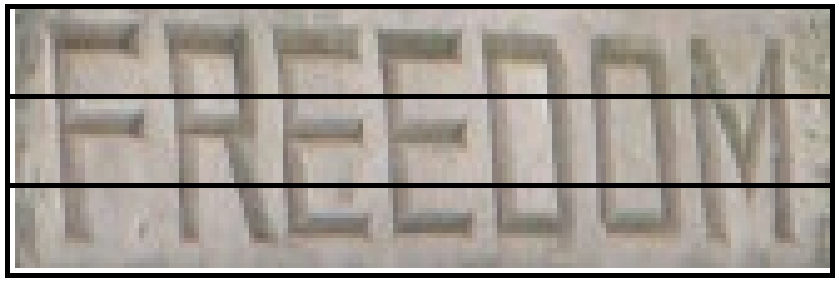
3x1



3hx1

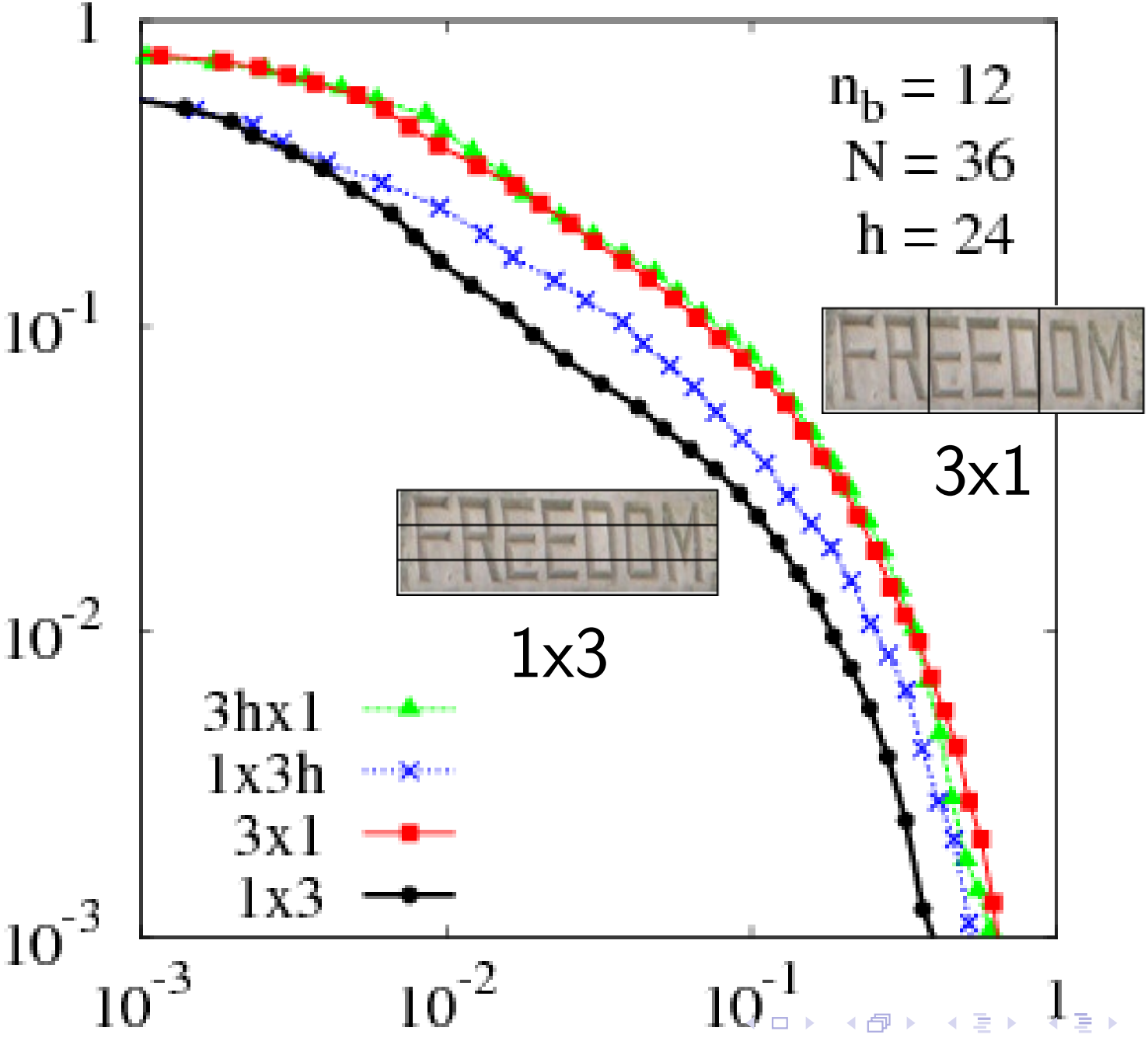


1x3h



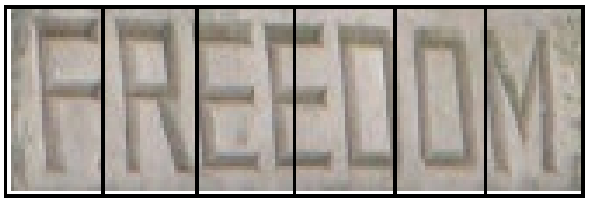
1x3

Decision error trade-off curves – 3 HOGs

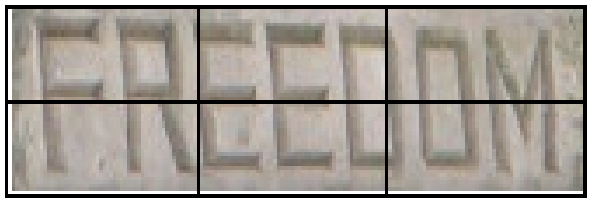


Standard HOG blocks/cells Arrangements

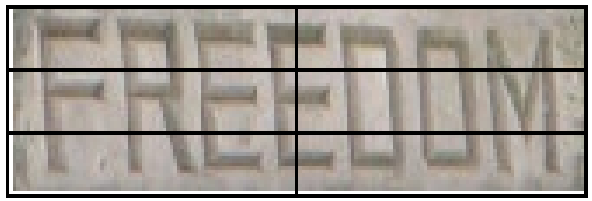
- Region splitting into 6 HOG's:



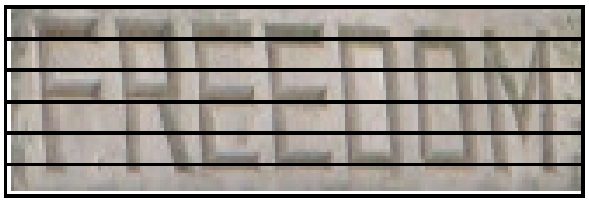
6x1



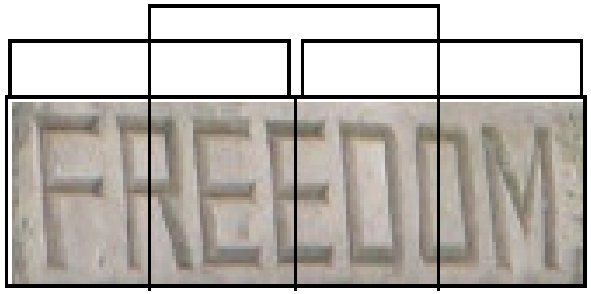
3x2



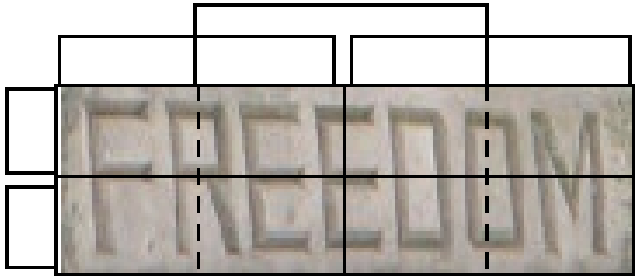
2x3



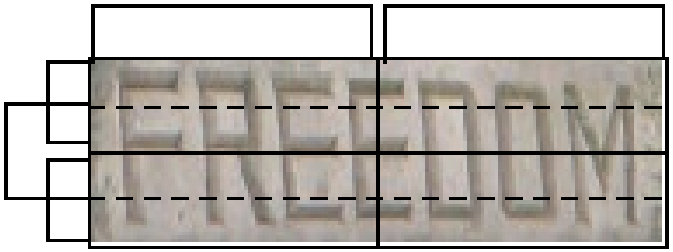
1x6



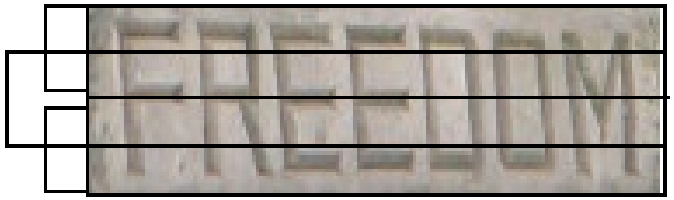
6fx1



3hx2

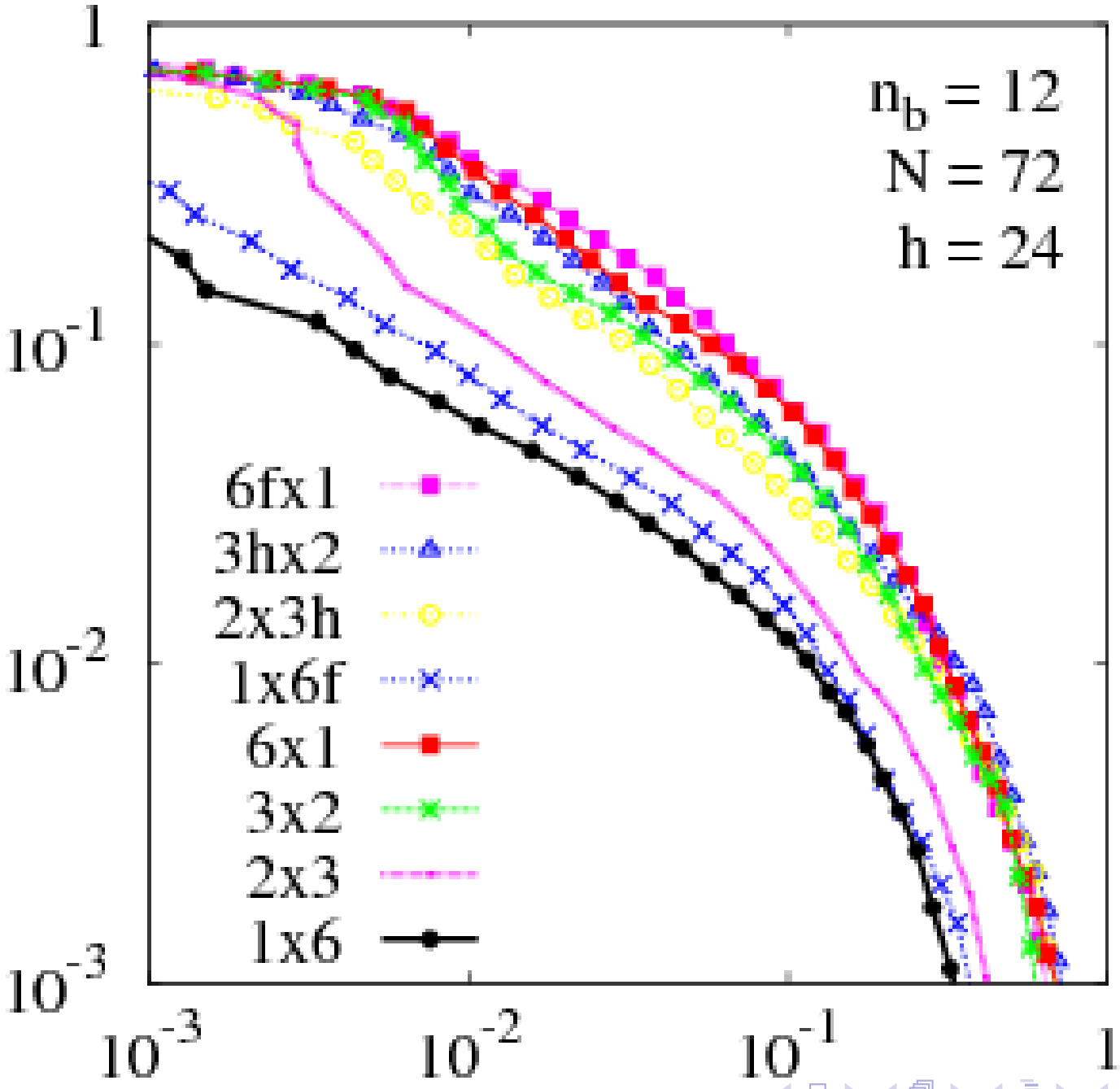


2x3h



1x6f

Decision error trade-off curves – 6 HOGs

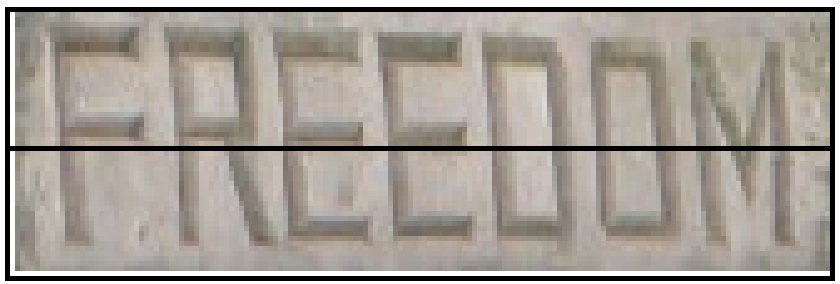


Standard HOG blocks/cells Arrangements

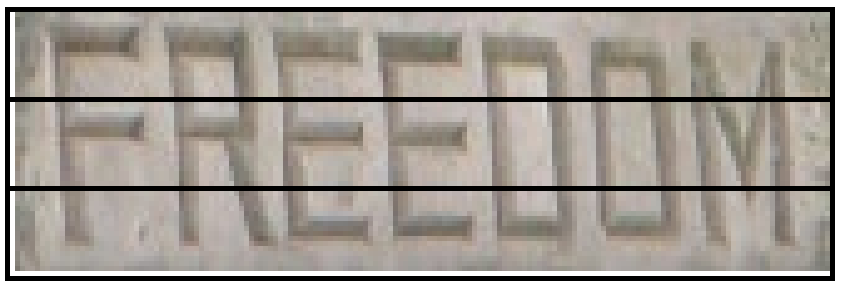
- **Conclusion**: horizontal stripes are better!

Standard HOG blocks/cells Arrangements

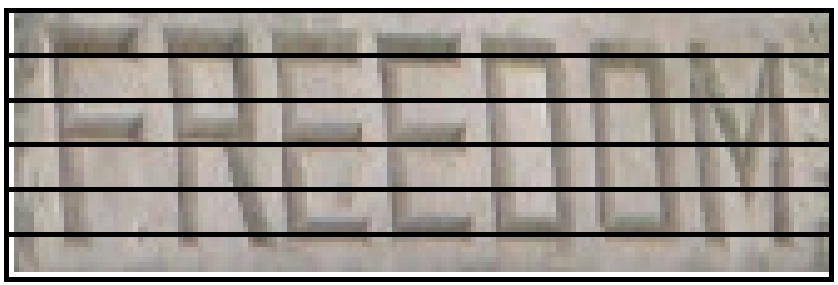
- **Conclusion**: horizontal stripes are better!
- What is the ideal number of horizontal stripes (n_y)?



1x2?

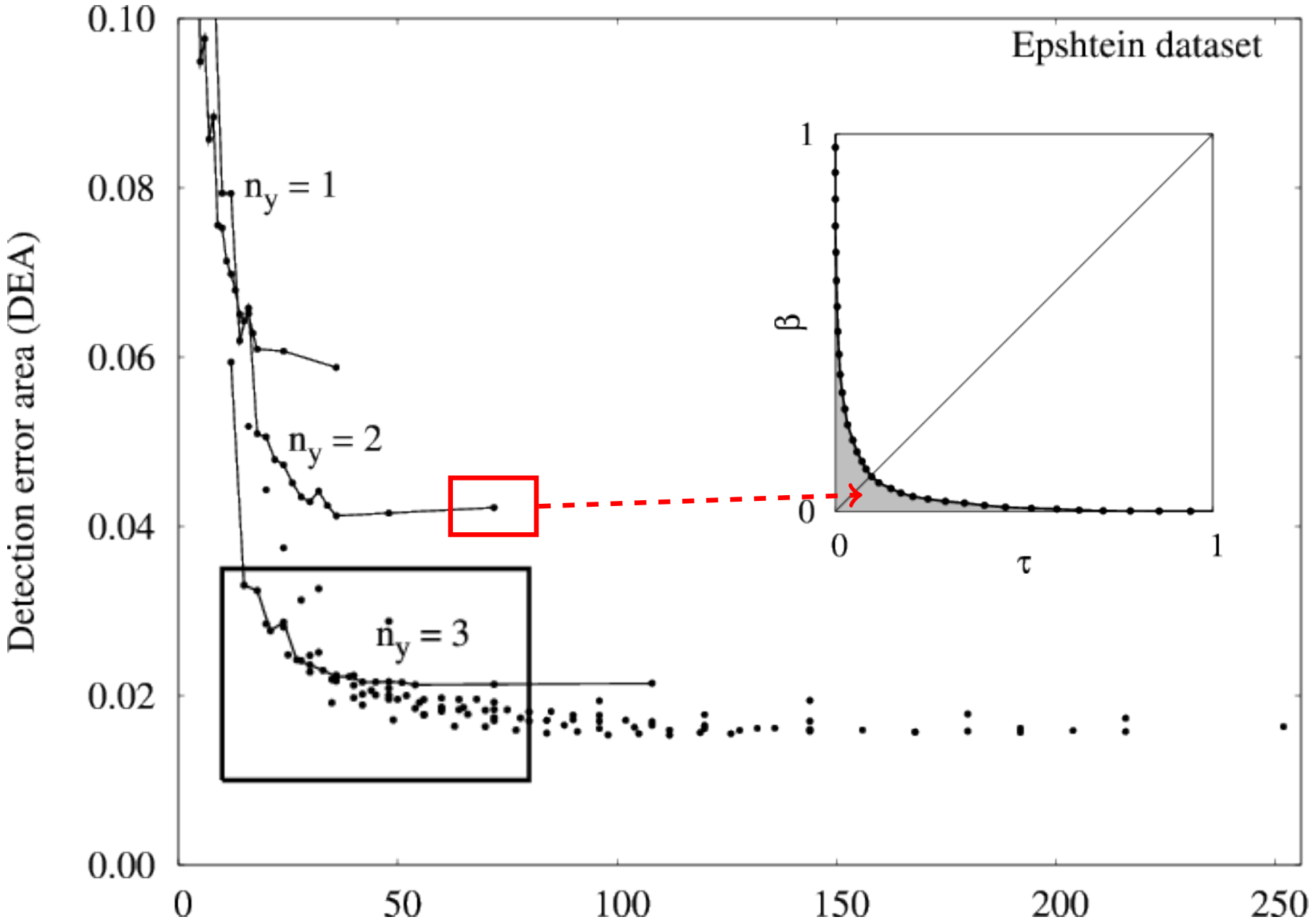


1x3?



1x6?

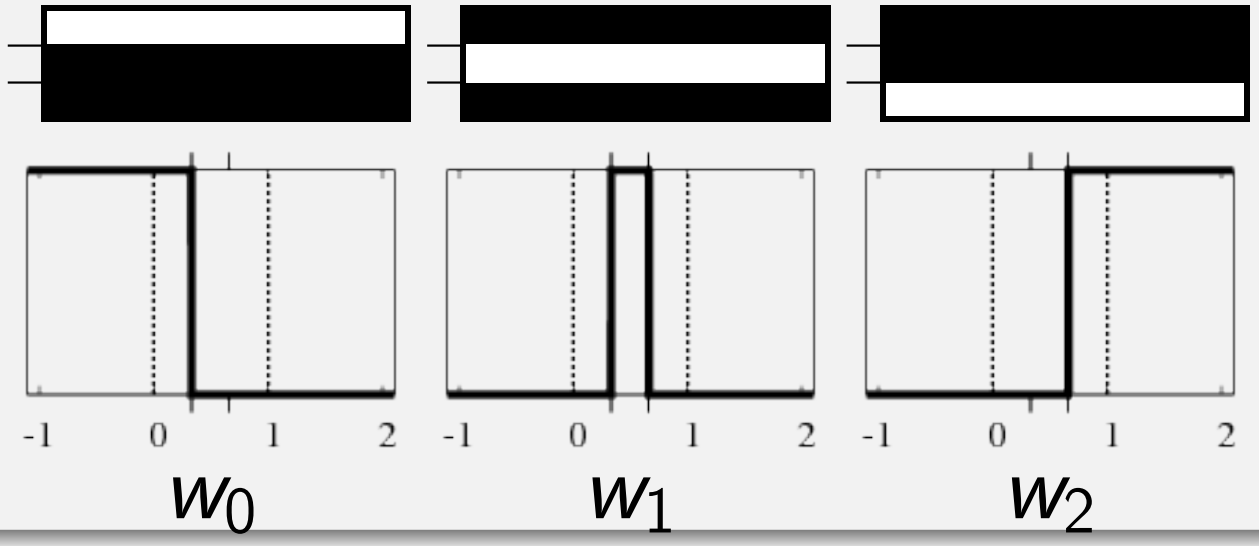
Standard HOG blocks/cells Arrangements

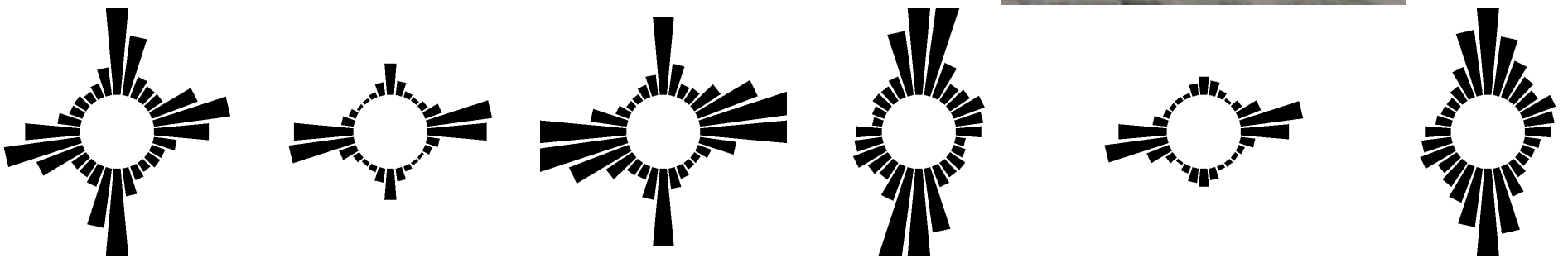


T-HOG cell weight functions

What is the importance of cell weight functions?

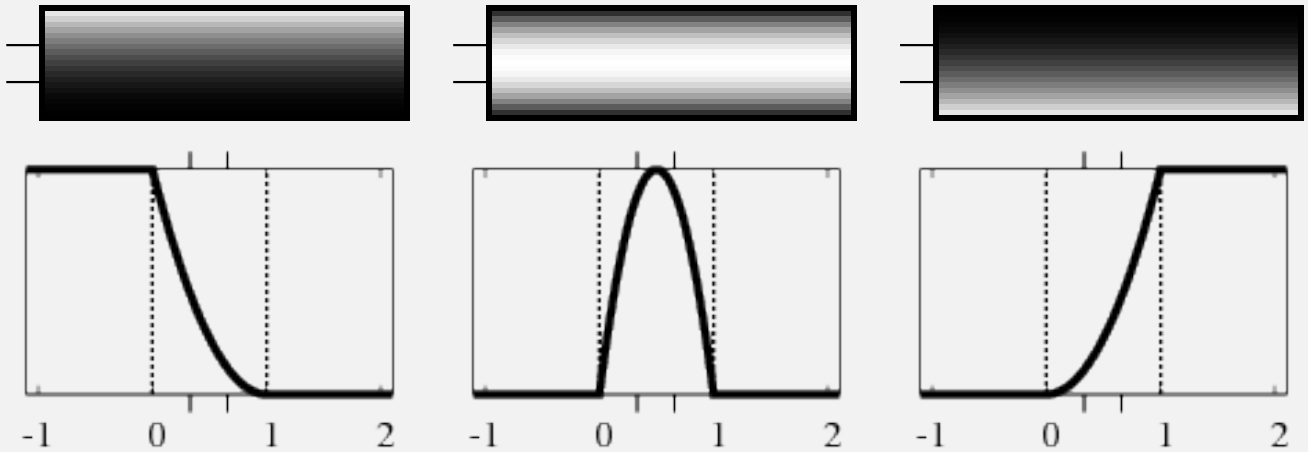
- Sharp cells:

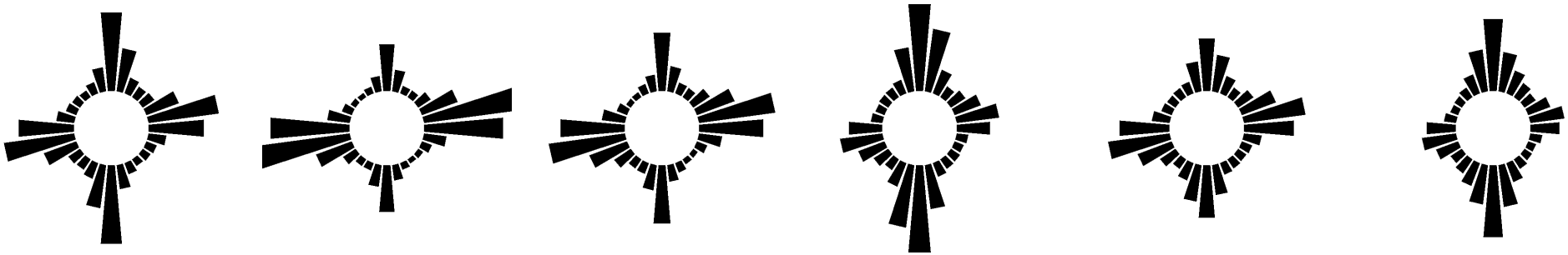




T-HOG cell weight functions

- Fuzzy cells:





Standard HOG cell weight functions

Problem: Sharp boundaries!

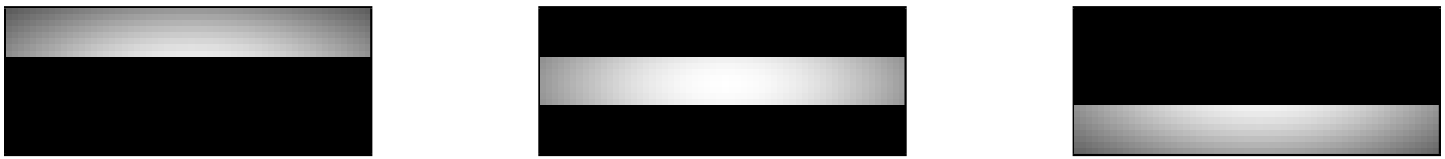


Figure: Single block: 1×3 cells ($\sigma/2$).

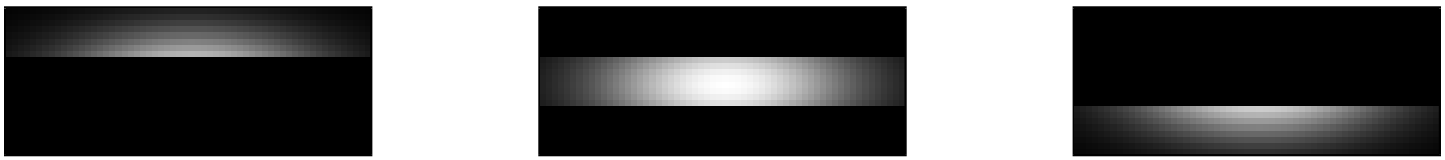


Figure: Single block of 1×3 cells ($\sigma/4$).

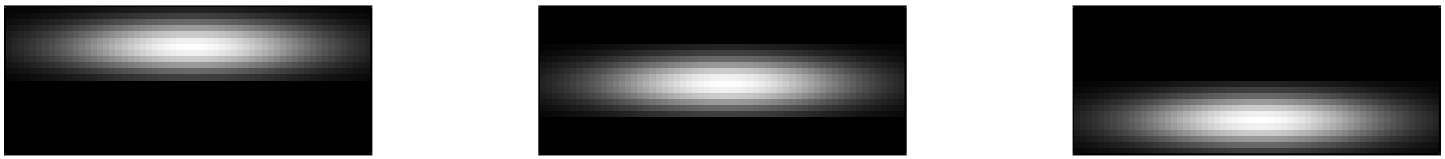
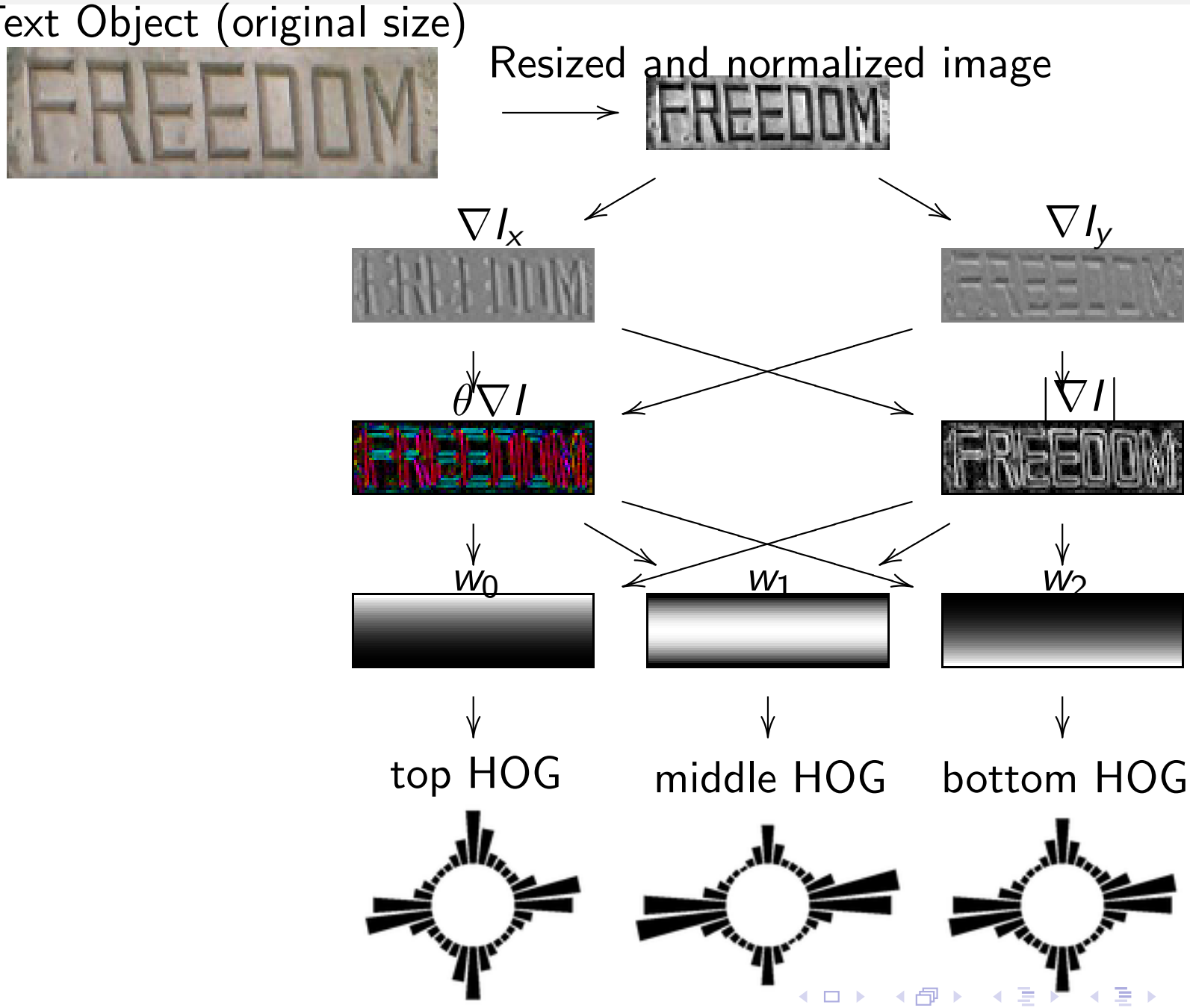


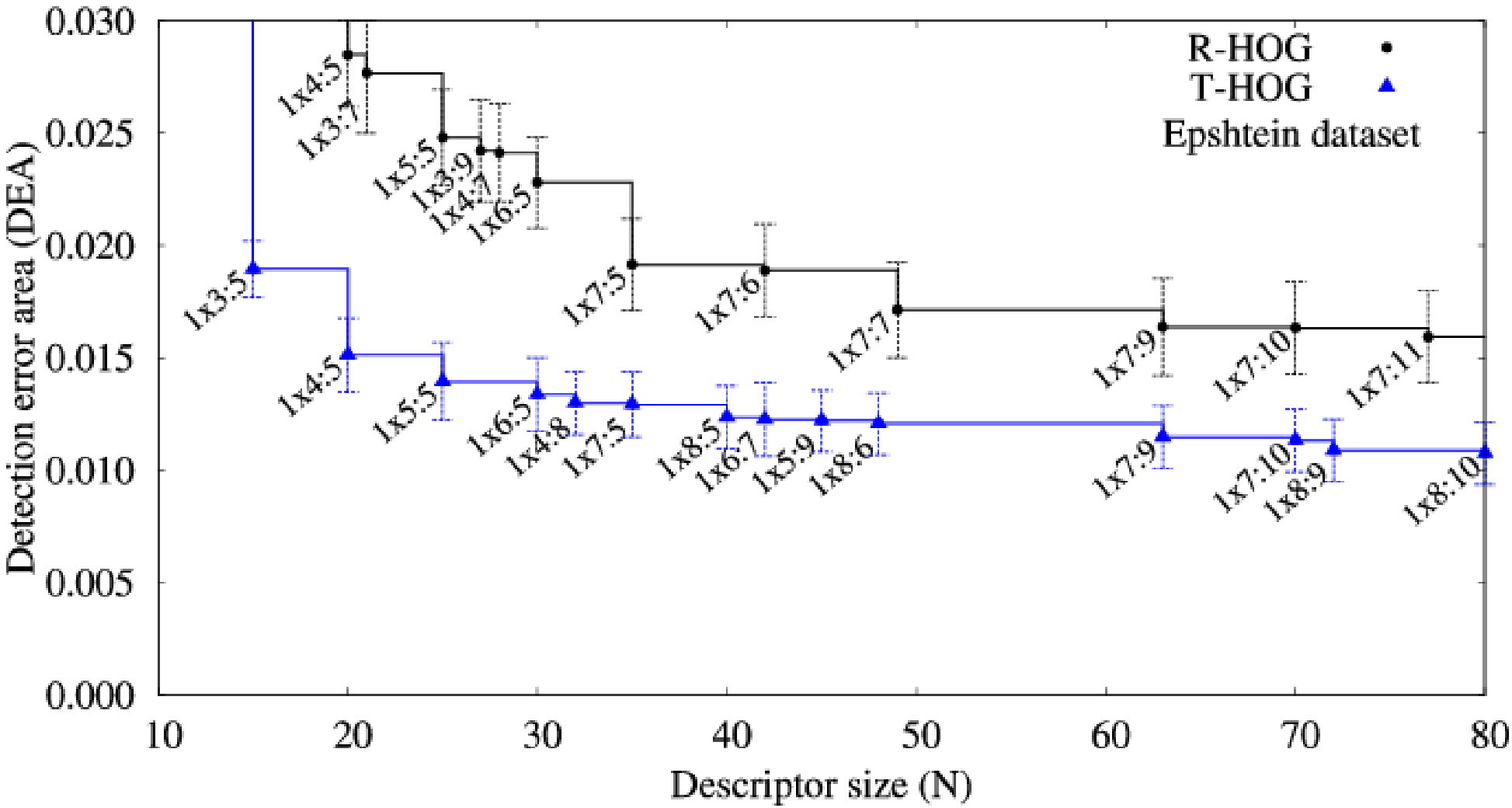
Figure: Overlapping 1×3 single-cell blocks.

T-HOG descriptor scheme



T-HOG x Standard HOG

Comparison T-HOG/R-HOG for text recognition:



SnooperText Scheme

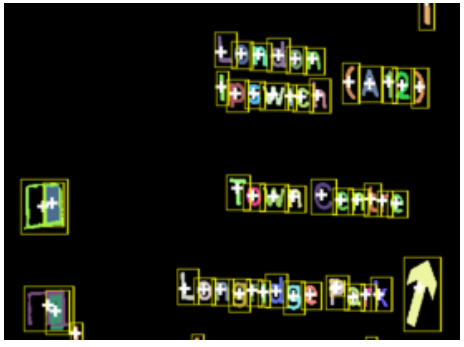
Segmentation Step

Refining Step

Build Pyramid

Input Image

Grouping Step



Letter Classification Step

Fourier descriptors

SVM

Pseudo Zernicke moments

SVM

Our polar descriptors

SVM

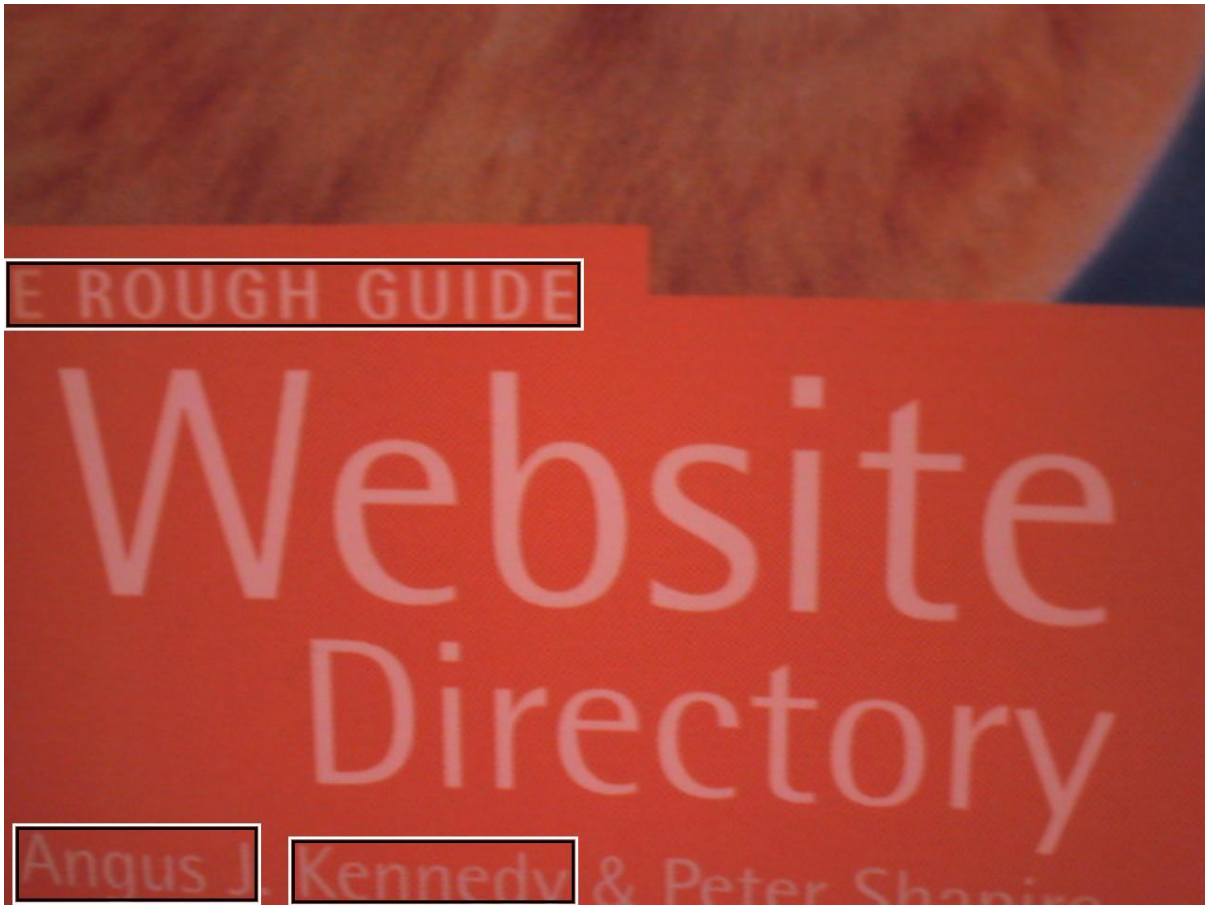
Final letter decision

SVM

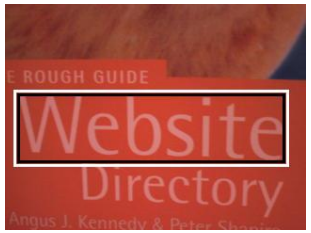
Experiments

SnooperText Multi-resolution:

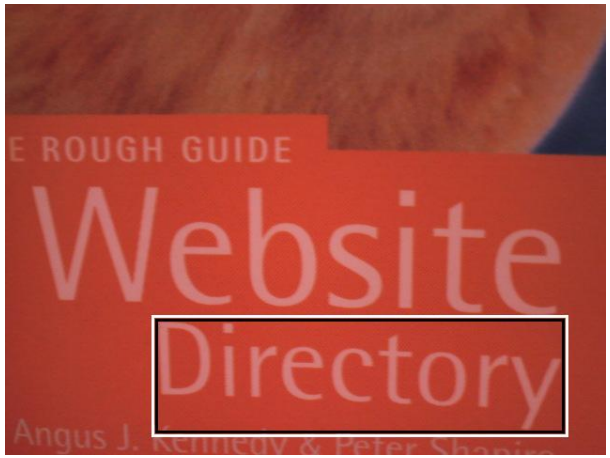
L^0



L^2

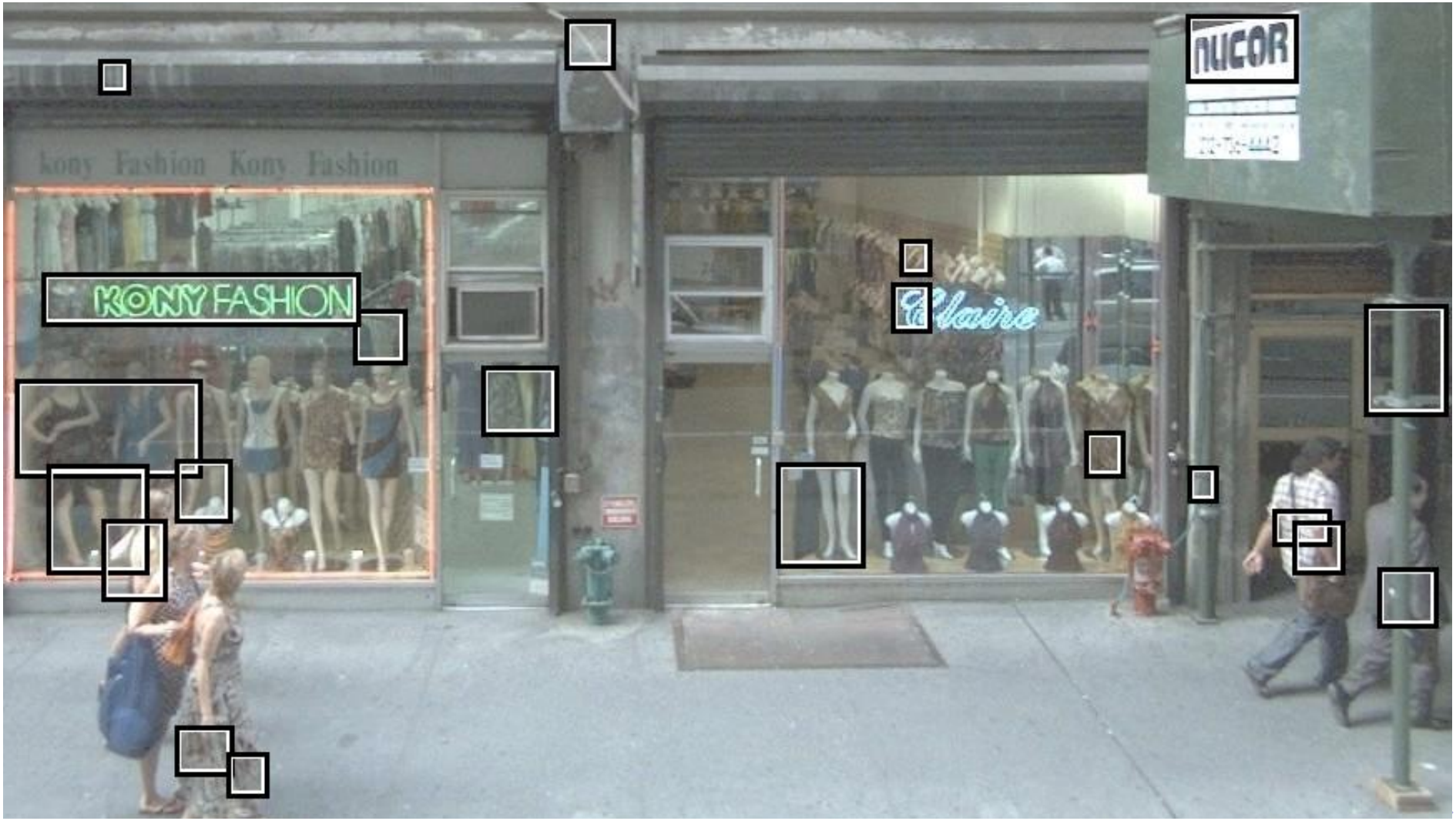


L^1



Experiments

SnooperText detection:



Experiments

SnooperText + T-HOG as a post-filter:

Experiments

Performance - ICDAR Challenge

System	p	r	f_{I}	f_{II}
SnooperText + T-HOG	0.73	0.61	0.65	0.67
Yi and Tian (TIP 2011)	0.71	0.62	0.62	0.66
H. Chen *et al.* (ICIP 2011)	0.73	0.60	—	0.66
Epshtein *et al.* (CVPR 2010)	0.73	0.60	—	0.66
Hinnerk Becker†	0.62	0.67	0.62	0.64
Alex Chen†	0.60	0.60	0.58	0.60
Ashida†	0.55	0.46	0.50	0.50
HWDavid†	0.44	0.46	0.45	0.45
Wolf†	0.30	0.44	0.35	0.36
Qiang Zhu†	0.33	0.40	0.33	0.36
...				

Experiments

Performance - Epshtein dataset

System	p	r	f_{I}	f_{II}
SnooperText + T-HOG	0.59	0.47	0.49	0.52
Epshtein *et al.* (CVPR 2010)	0.54	0.42	—	0.47

Performance - iTowns dataset

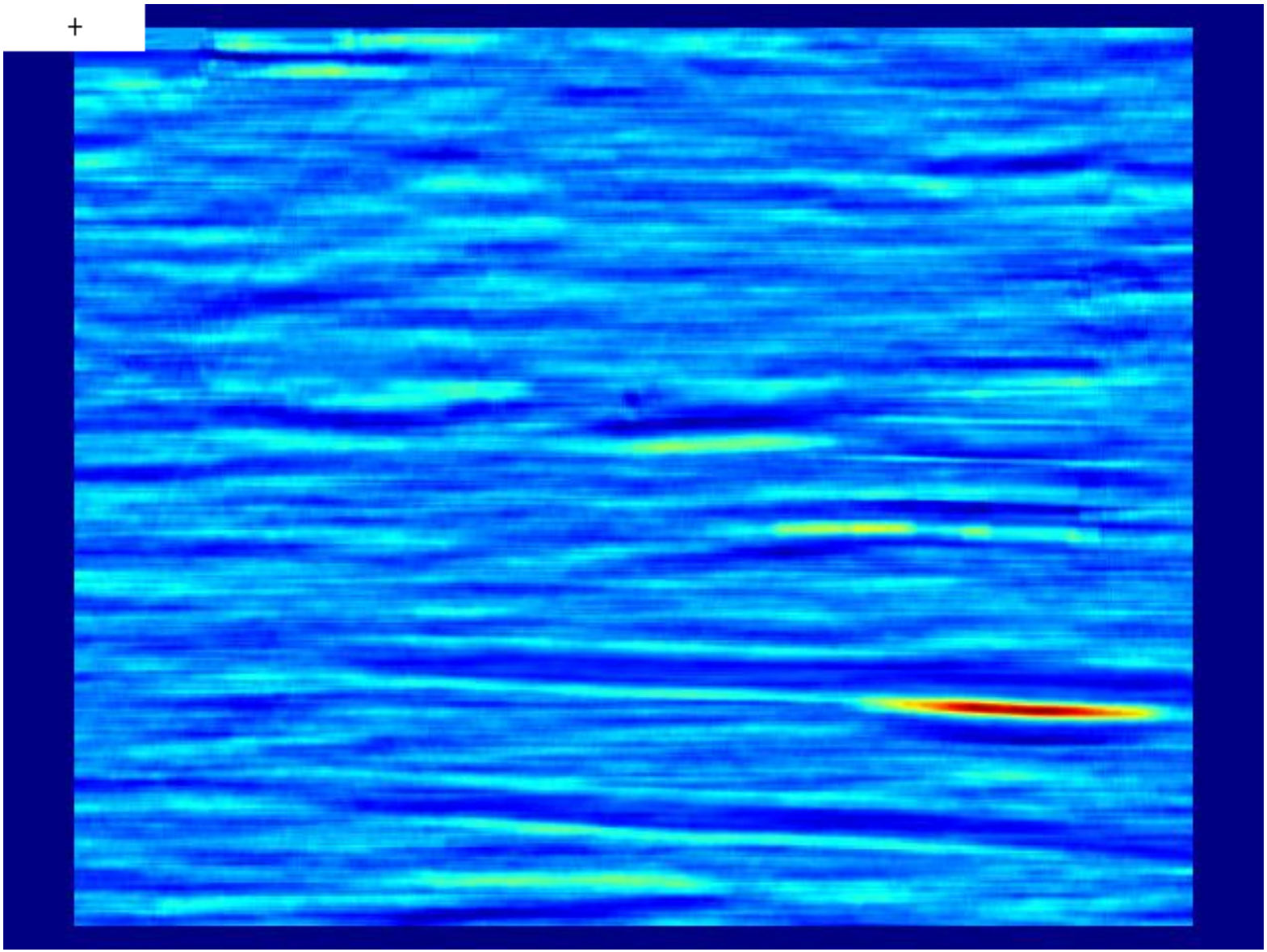
System	p	r	f_{I}	f_{II}
SnooperText + T-HOG	0.72	0.50	0.56	0.59

Experiments

T-HOG as sliding window text detector:

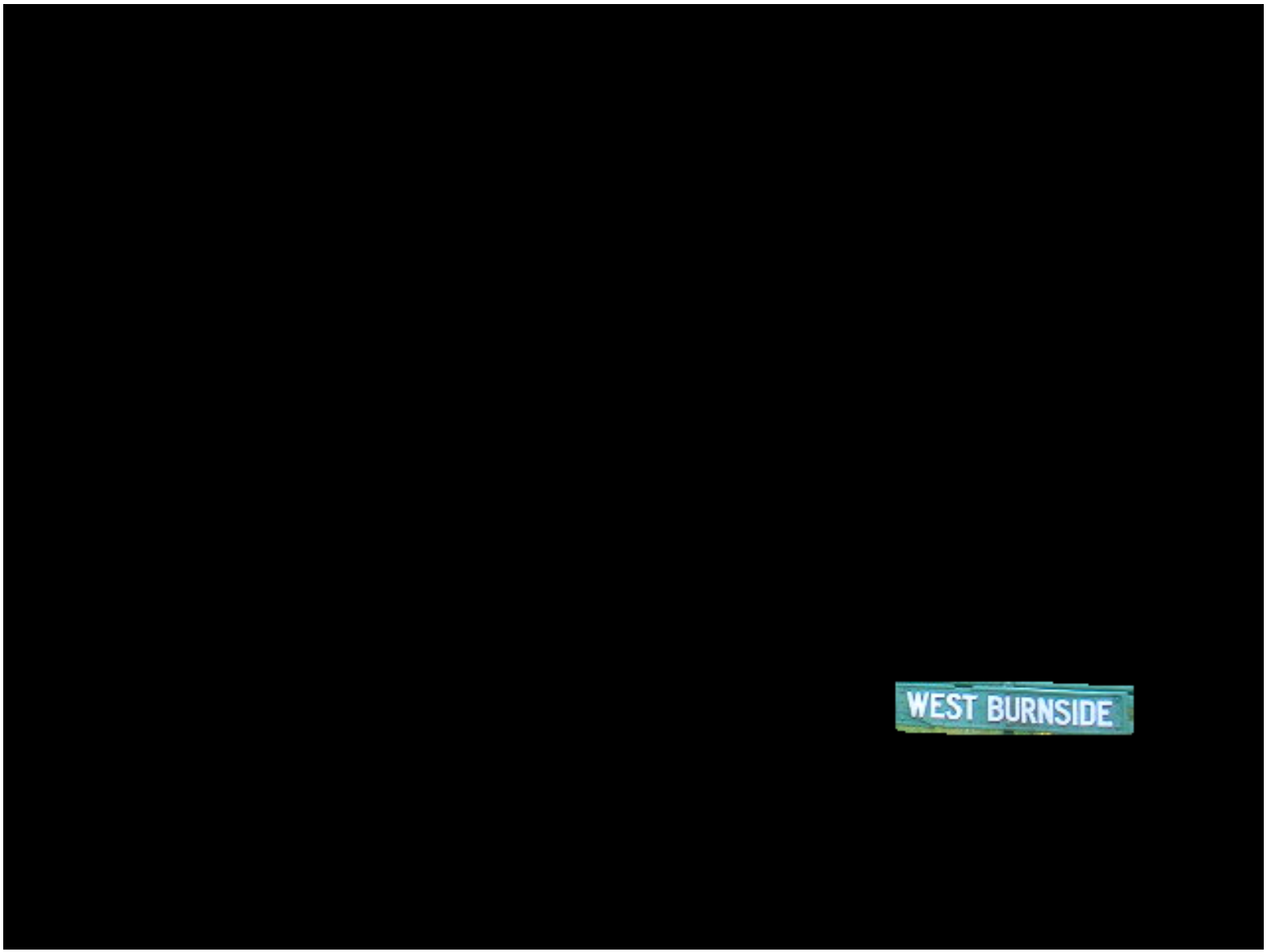
Experiments

T-HOG as sliding window text detector:



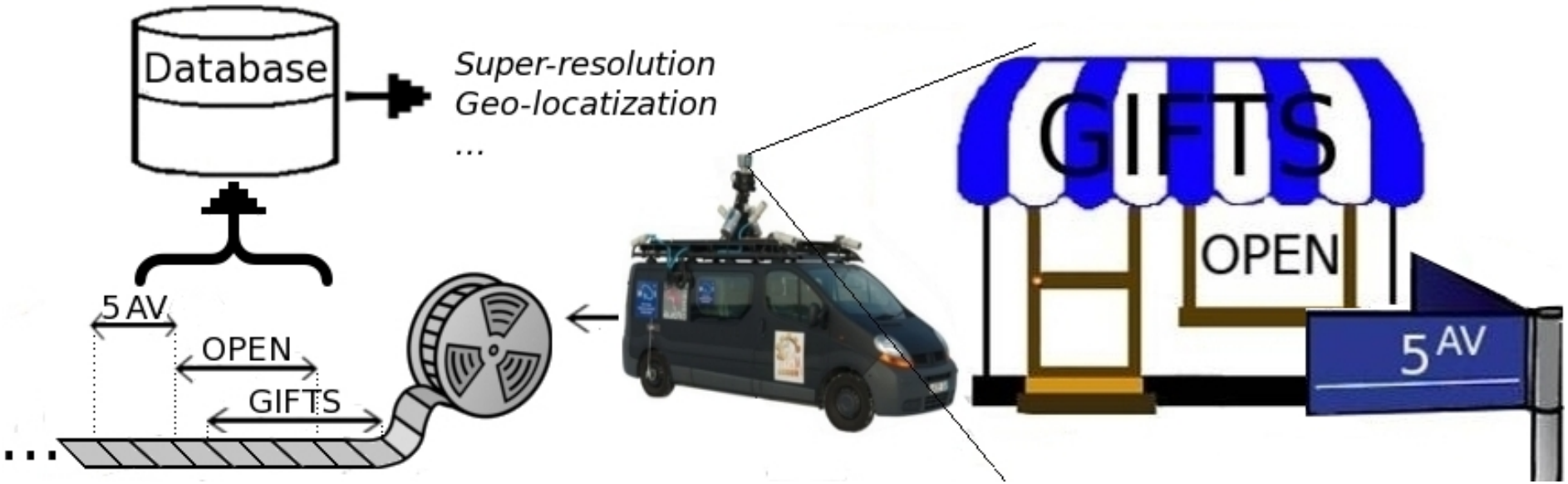
Experiments

T-HOG as sliding window text detector:



Part II

Text tracking



Contributions

- Description of 4 text tracking strategies:
  - SnooperTrack.
- Definition of special metrics for text tracking systems;
- Description of a PF designed specifically for text:
  - T-HOG: contents similarity;
  - T-HOG: contents classification.
- Benchmark with 6 videos of urban scenes (XML).

Statement of the problem

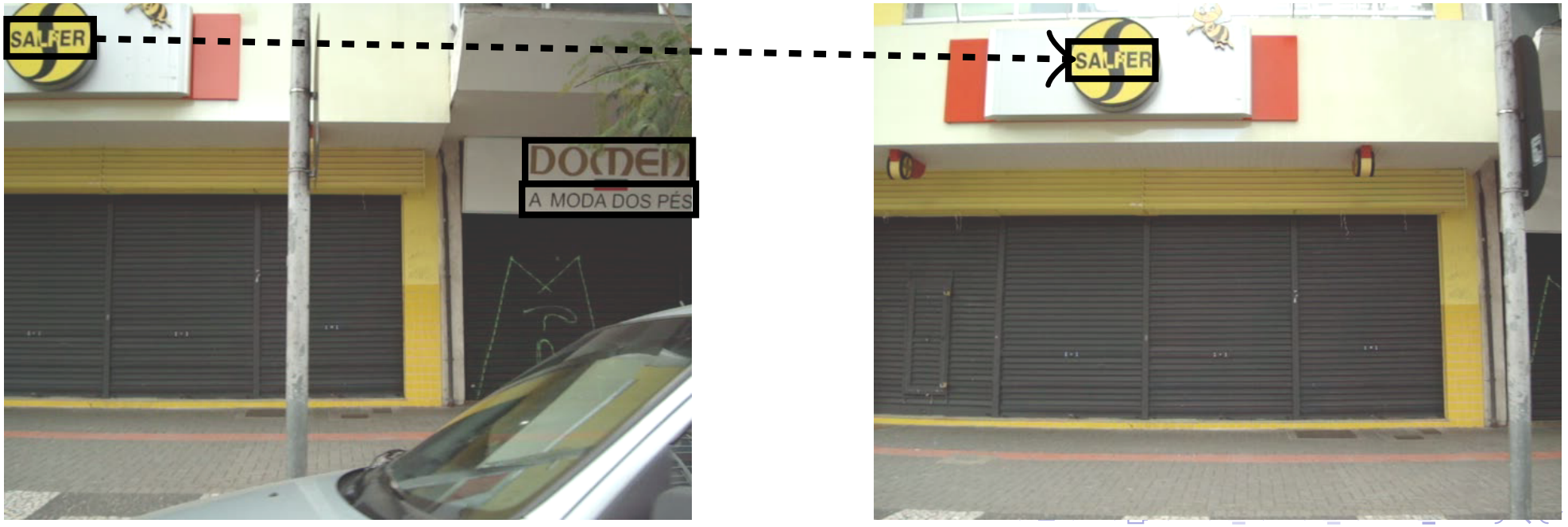
- Input data: a video $\mathbb{V} \longrightarrow$ frames $\mathbb{V}^{(0)}, \mathbb{V}^{(1)}, \ldots, \mathbb{V}^{(n-1)}$;
- Text detection;
- Text tracking;

$\mathbb{V}^{(i-2)}$ $\mathbb{V}^{(i-1)}$

Statement of the problem

- Input data: a video $\mathbb{V} \longrightarrow$ frames $\mathbb{V}^{(0)}, \mathbb{V}^{(1)}, \ldots, \mathbb{V}^{(n-1)}$;
- Text detection;
- Text tracking;

$\mathbb{V}^{(i-1)}$ $\mathbb{V}^{(i)}$



Statement of the problem

- Output data:

$$\ldots \quad \mathbb{V}^{(i-2)} \qquad \mathbb{V}^{(i-1)} \qquad \mathbb{V}^{(i)} \quad \ldots$$

Statement of the problem

- Output data:
  - List of text regions $T^{(0)}, T^{(1)}, \ldots, T^{(n-1)}$;

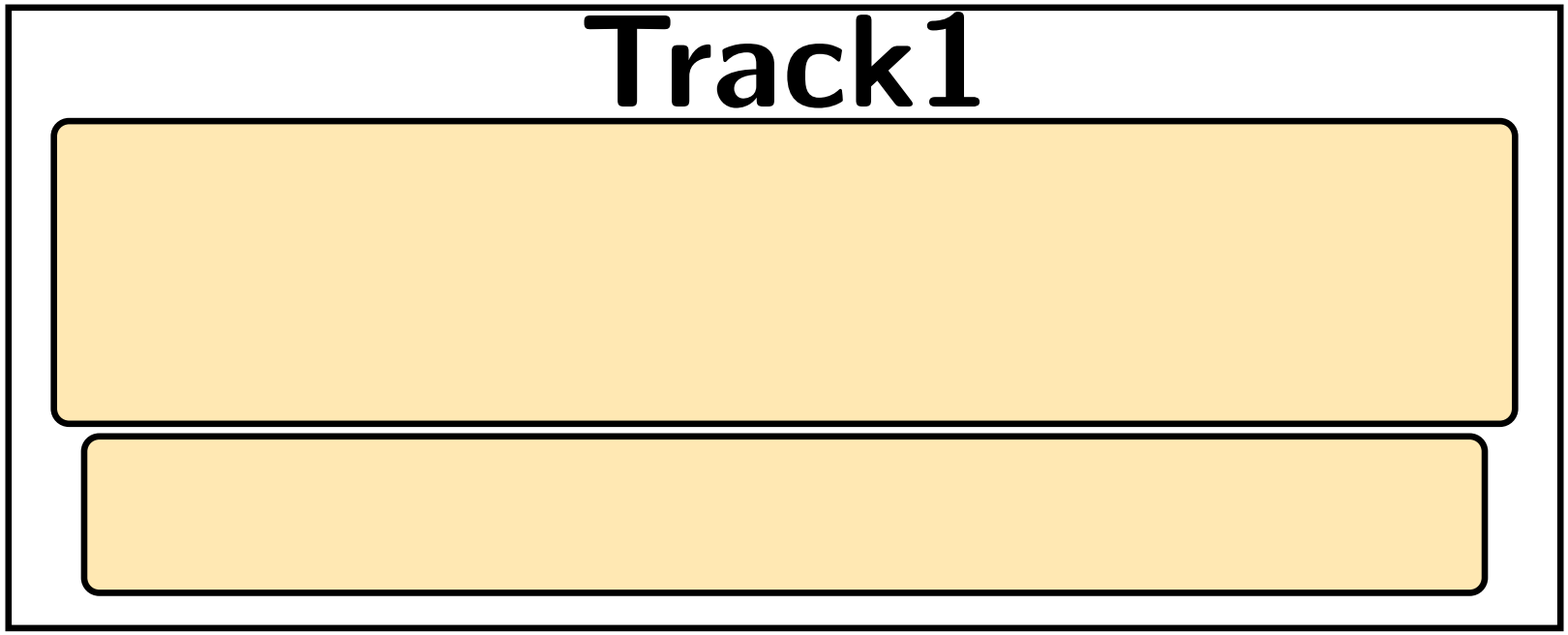
$\ldots \quad \mathbb{V}^{(i-2)} \qquad \mathbb{V}^{(i-1)} \qquad \mathbb{V}^{(i)} \quad \ldots$

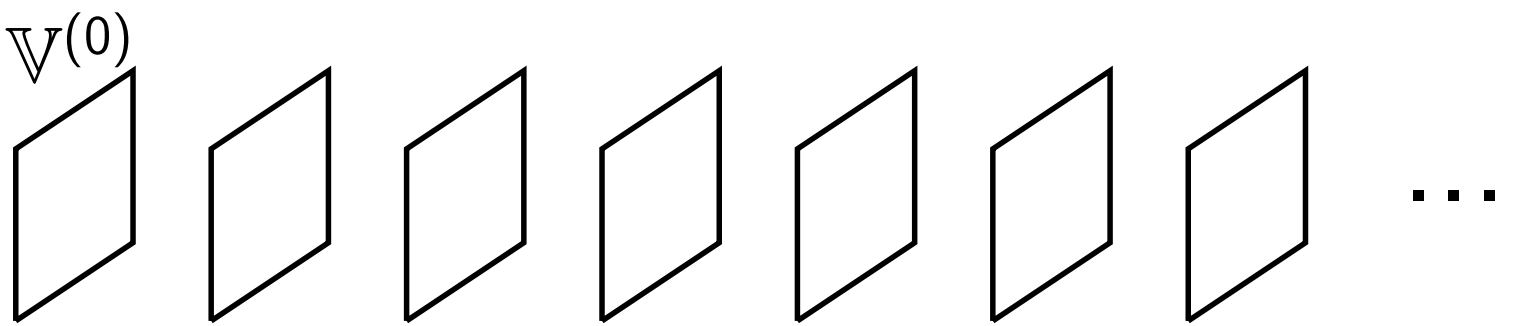
Statement of the problem

- Output data:
  - List of text regions $T^{(0)}, T^{(1)}, \ldots, T^{(n-1)}$;
  - Tracking relations $\pi^{(1)}, \pi^{(2)}, \ldots, \pi^{(n-1)}$;

$$\ldots \quad \mathbb{V}^{(i-2)} \qquad \mathbb{V}^{(i-1)} \qquad \mathbb{V}^{(i)} \quad \ldots$$

Tracking Strategies

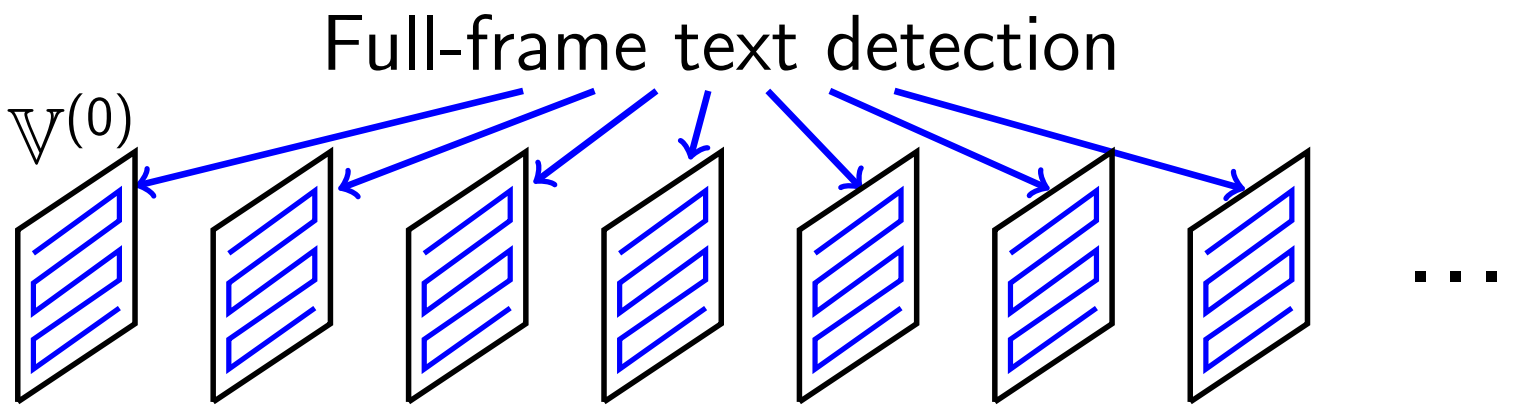




Tracking Strategies

Track1

Full-frame text detection on every frame

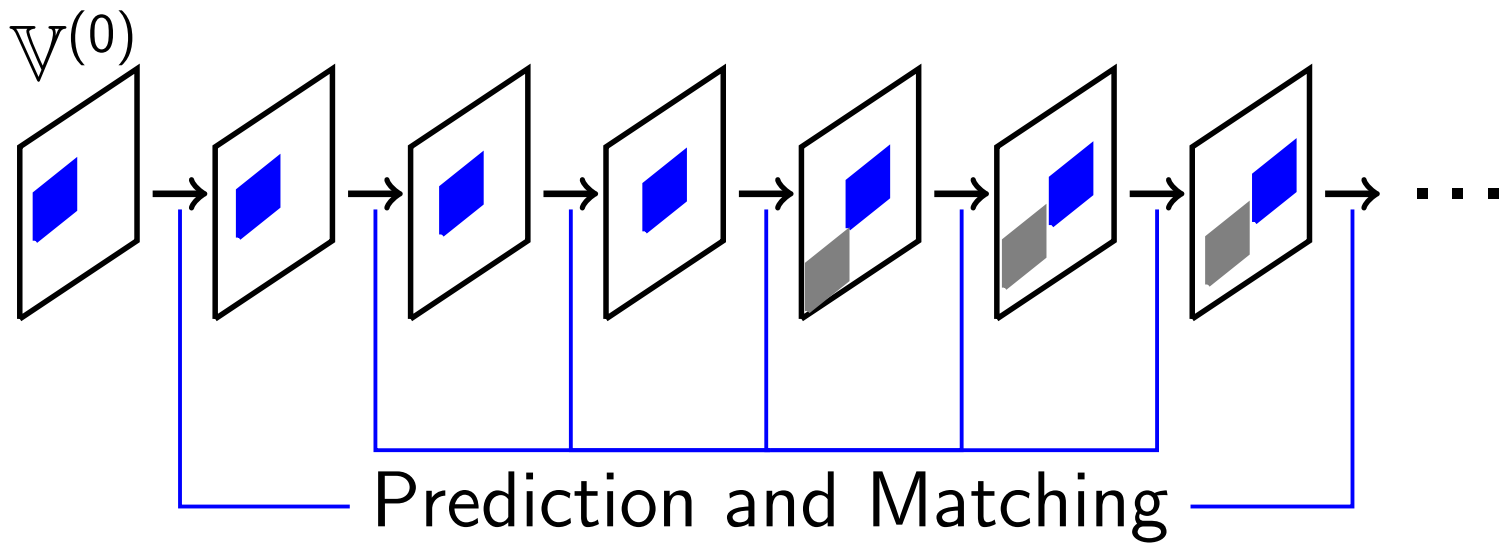


Tracking Strategies

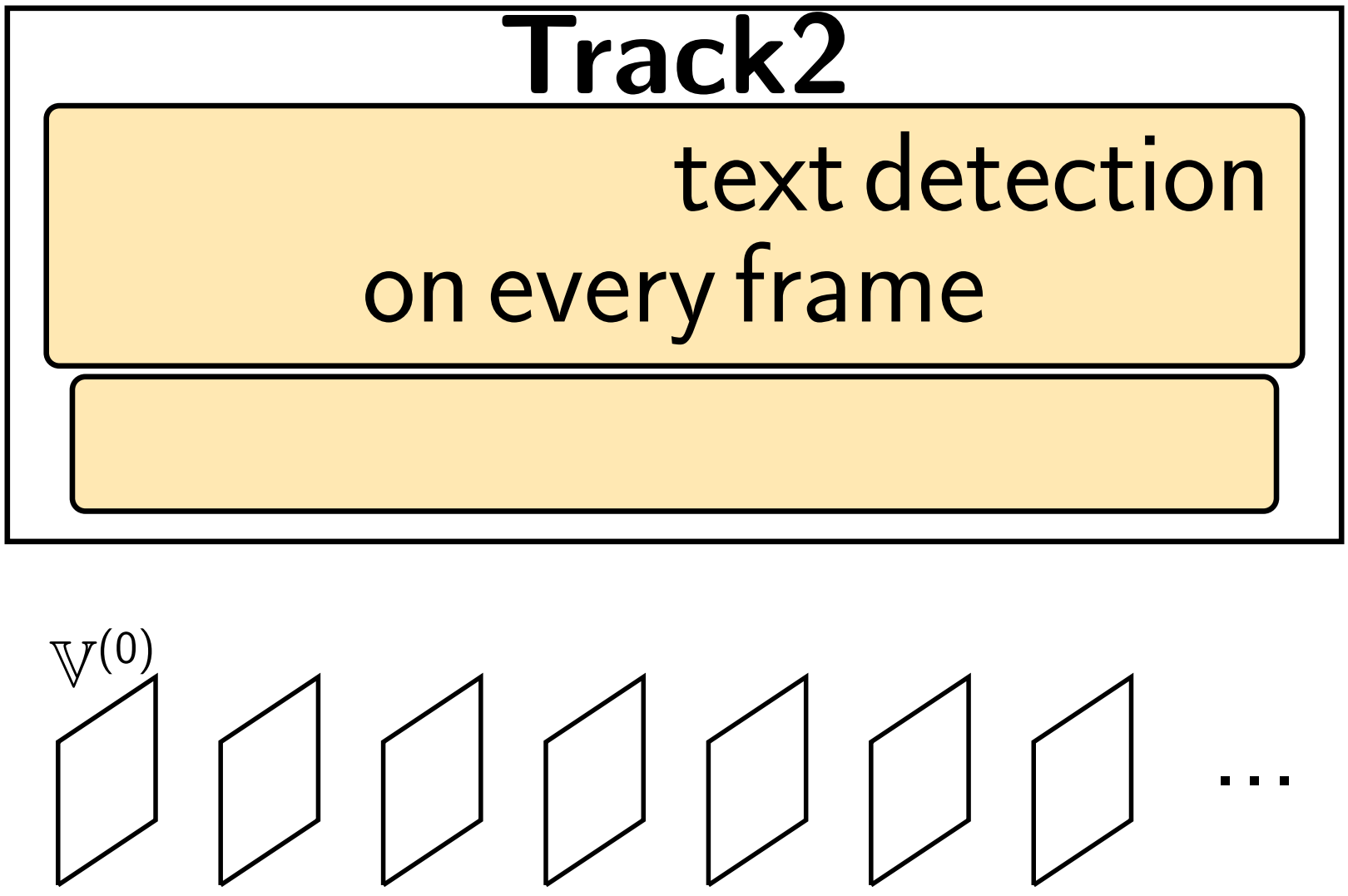
Track1

Full-frame text detection on every frame

Prediction and Matching



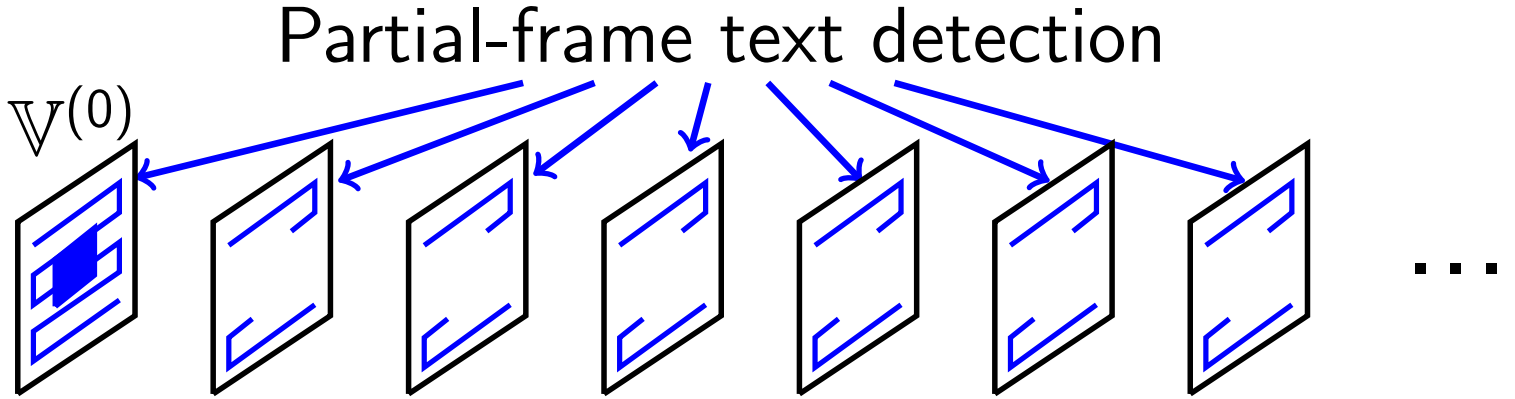
Tracking Strategies



Tracking Strategies

Track2

Partial-frame text detection on every frame

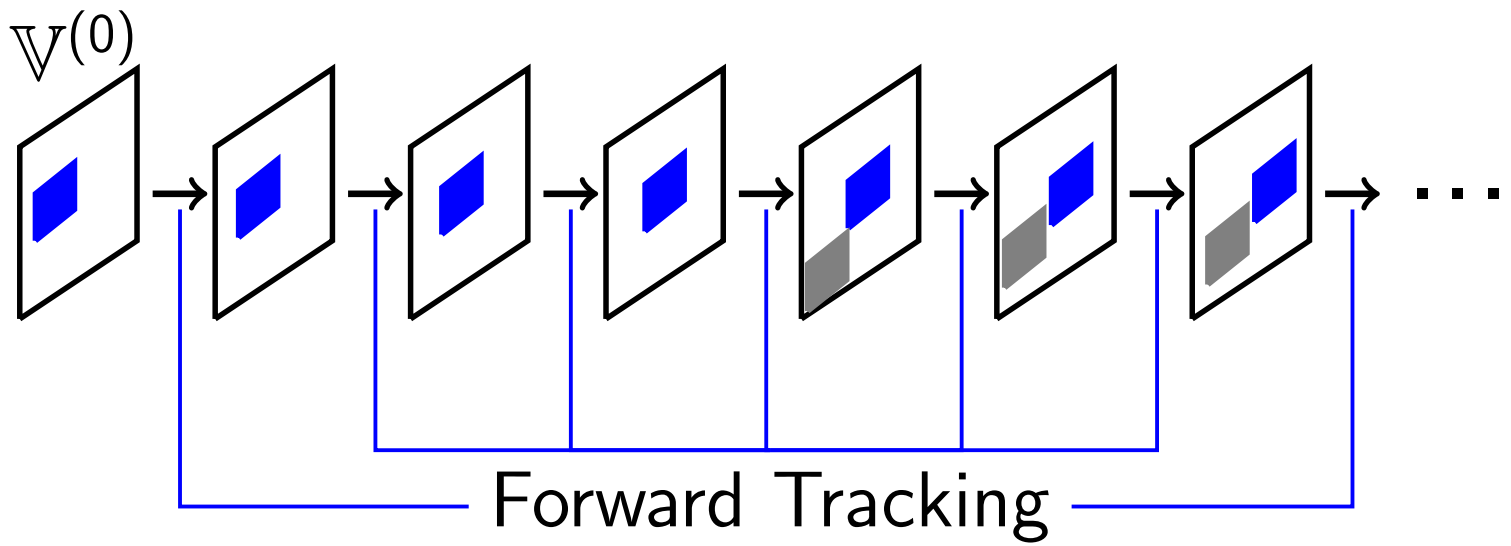


Tracking Strategies

Track2

Partial-frame text detection on every frame

Forward Tracking

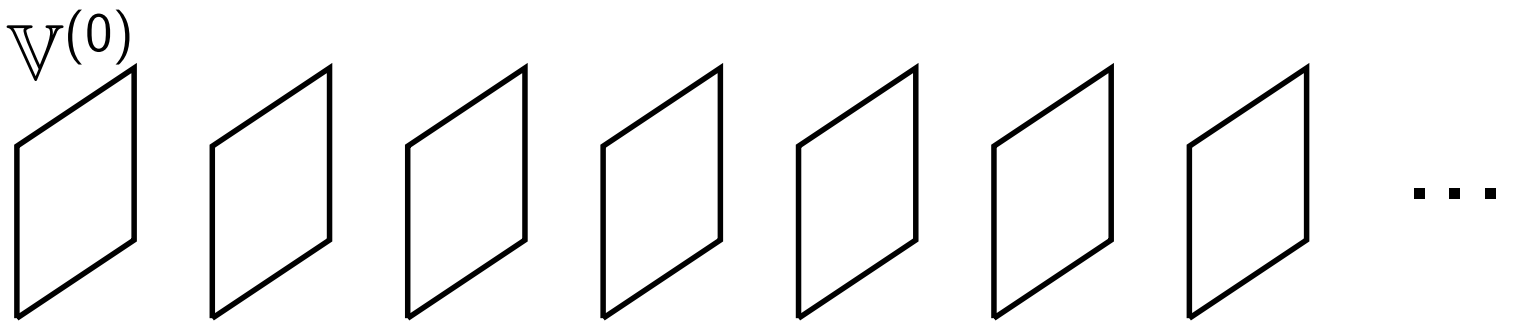


Tracking Strategies

Track3 - SnooperTrack

Partial-frame text detection

Forward Tracking

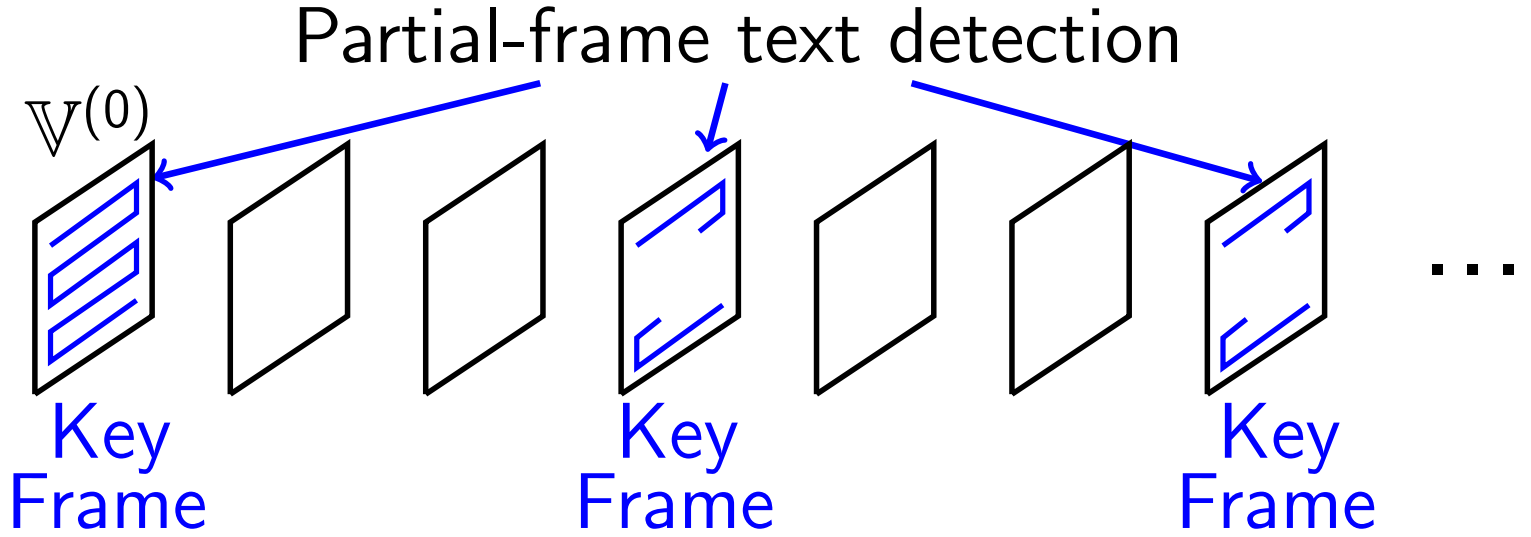


Tracking Strategies

Track3 - SnooperTrack

Partial-frame text detection on key frames

Forward Tracking

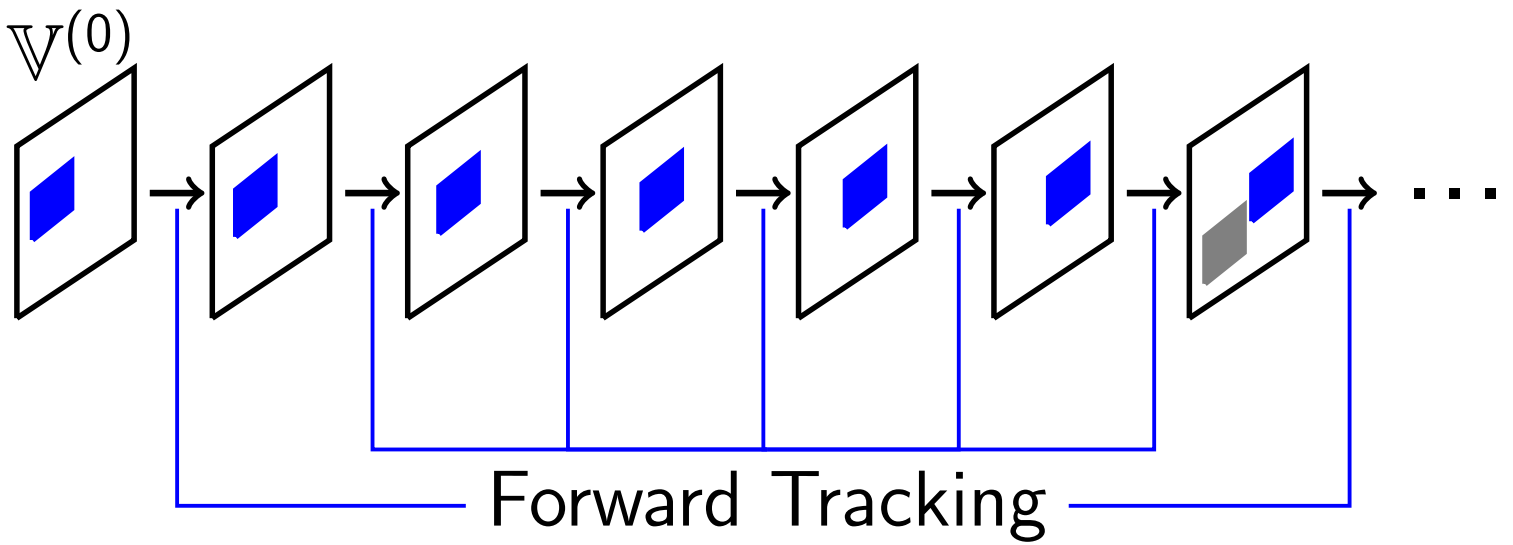


Tracking Strategies

Track3 - SnooperTrack

Partial-frame text detection on key frames

Forward Tracking

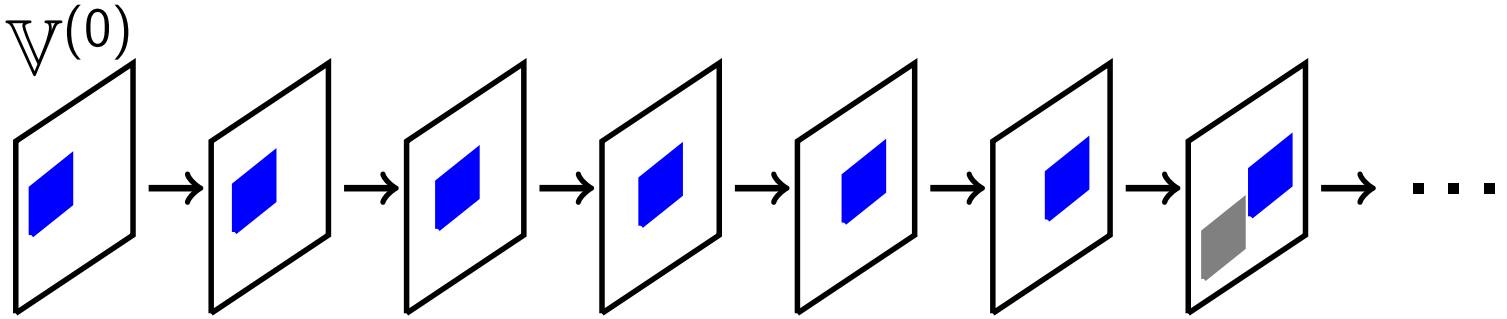


Tracking Strategies

Track4

Partial-frame text detection on key frames

Tracking

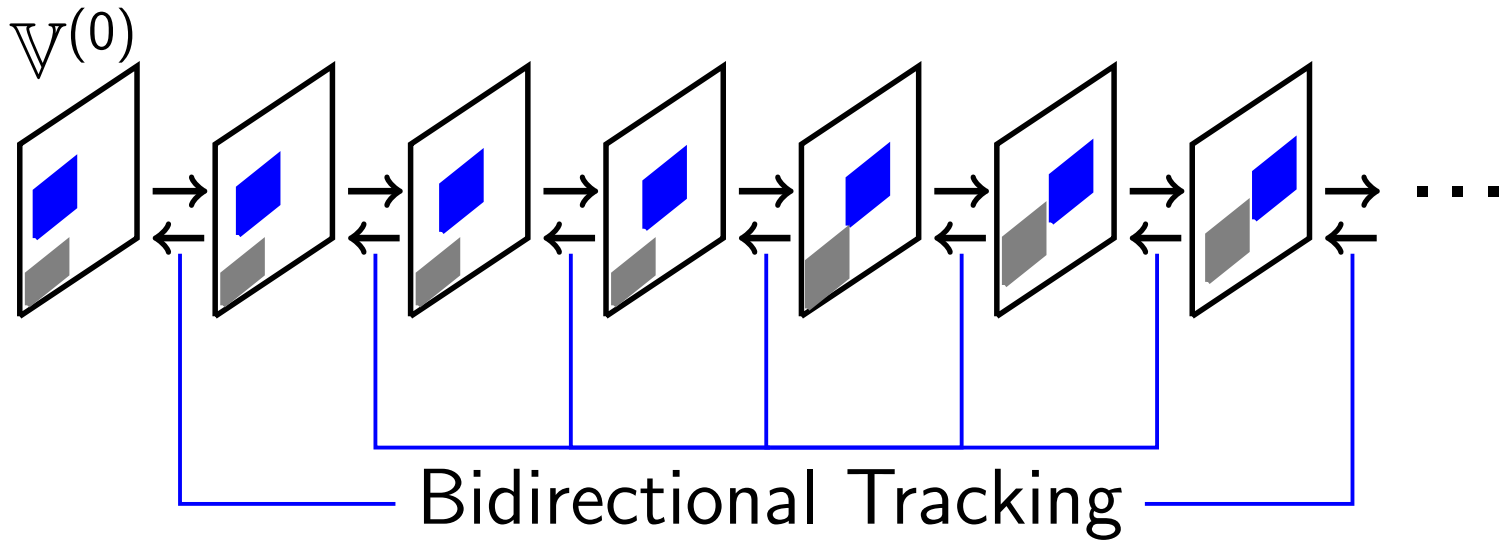


Tracking Strategies

Track4

Partial-frame text detection on key frames

Bidirectional Tracking



Experiments

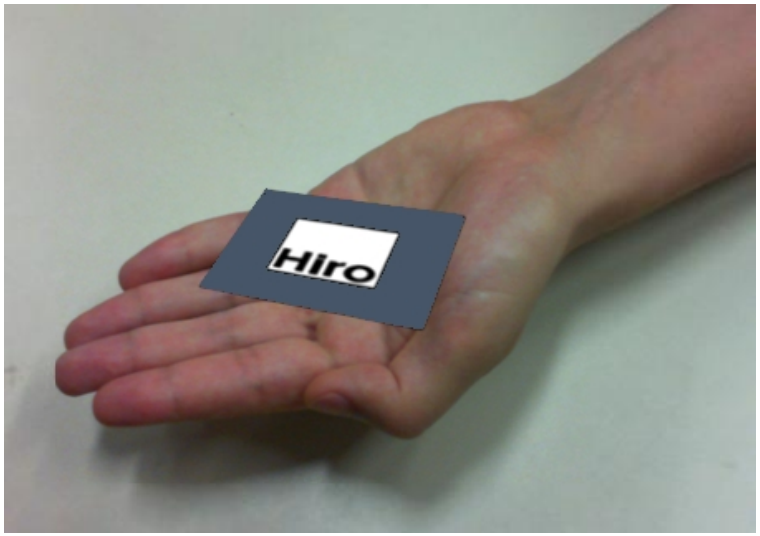
- Video dataset (6 outdoor videos);
- Metrics:
  - Region detection accuracy;
  - Object detection accuracy;
  - Text tracking accuracy;
  - ICDAR detection score;
- Text detector: SnooperText/Ideal.
- Text tracker:
  - T1 (simple linear prediction);
  - T2–4 (particle filter + T-HOG).

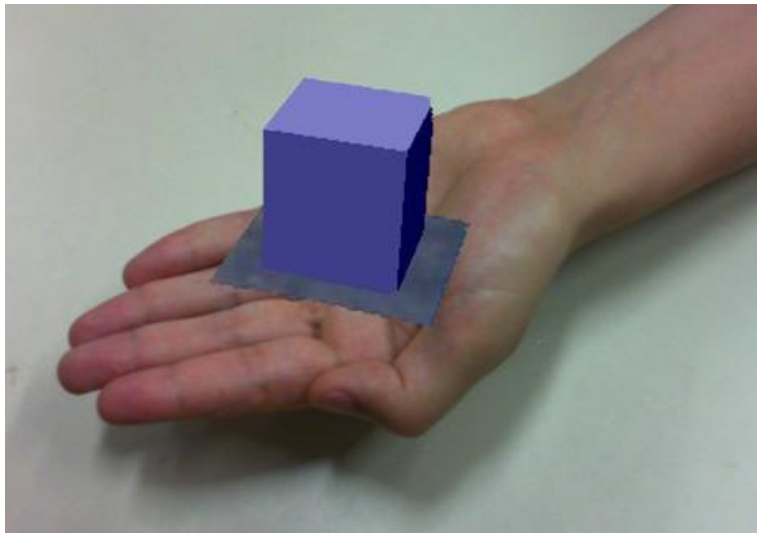
Conclusions

- Impact of text detection errors;
- Impact of false positive detections;
- Impact of over-segmentation:
- Impact of backward tracking:
  - $\uparrow$ f-score improved 4% in all metrics.
- Track1 with an ideal text detector was the best;
- Track3(ST)–Track4 with a real text detector were the best;
- Track1–Track2 are more time consuming than Trak3–Track4. All algorithms using the SnooperText detector.

Part III

Tracking of 3D objects





Contributions

- Development of a new algorithm – AFFTrack:
  - Camera calibration;
  - Feature tracking;
- Development of a benchmark (publicly available).

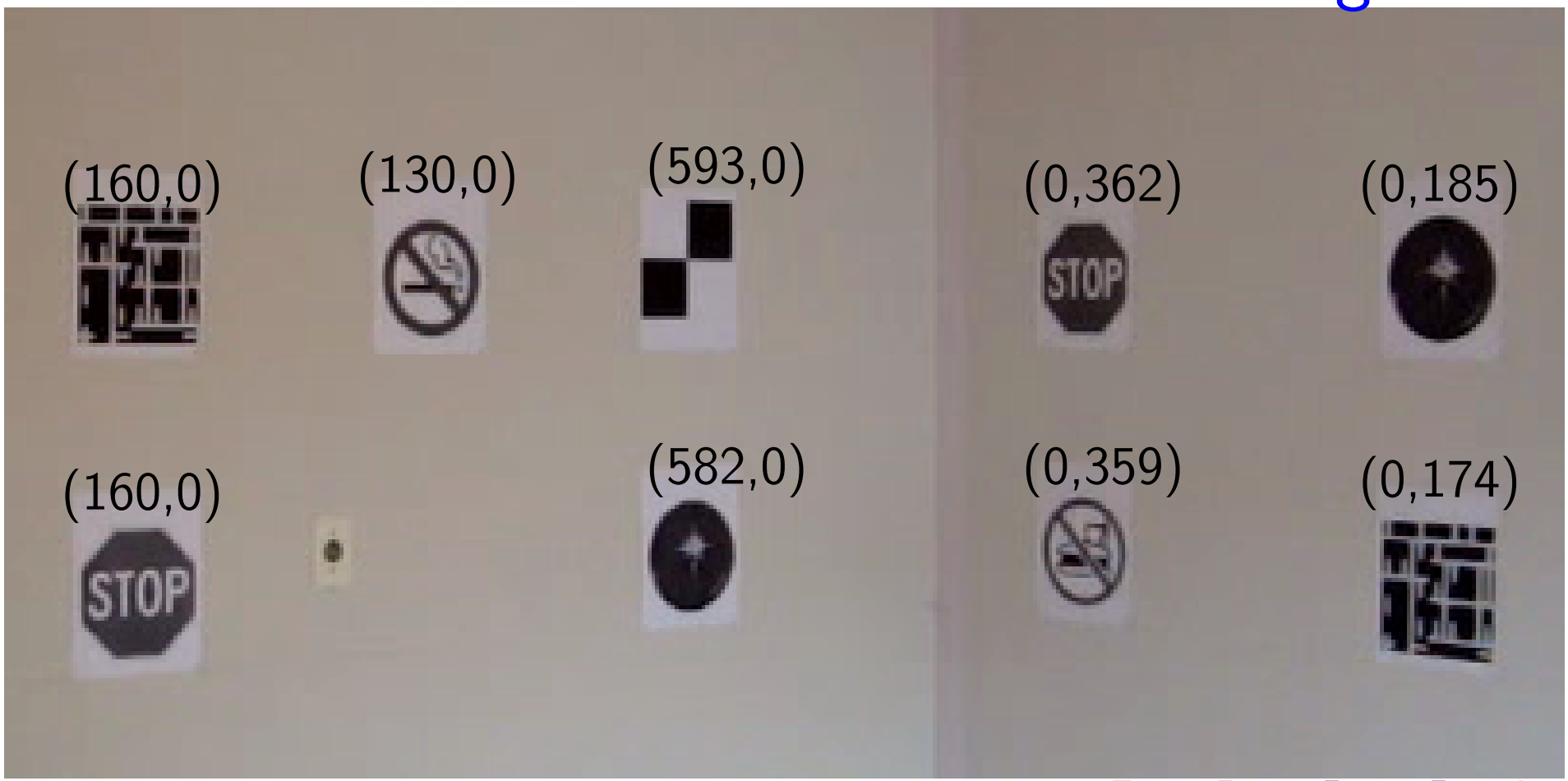
Statement of the problem

Input:

$\longrightarrow$ A video $\mathbb{V}$ $(\mathbb{V}^{(0)}, \mathbb{V}^{(1)}, \ldots, \mathbb{V}^{(n-1)})$;

$\longrightarrow$ Initial frame positions at $\mathbb{V}^{(0)}$.

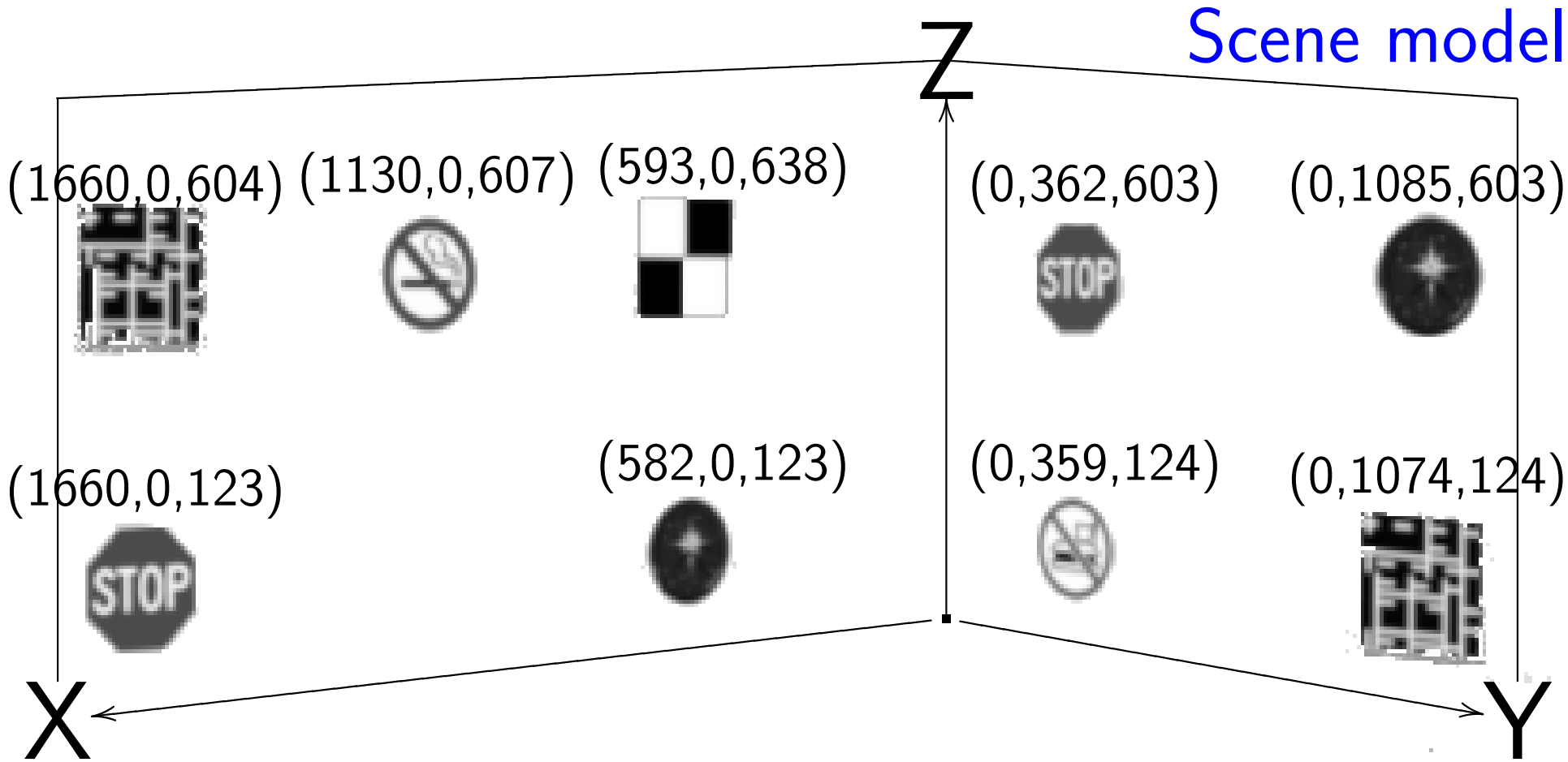
Image $\mathbb{V}^{(0)}$



Statement of the problem

Input:

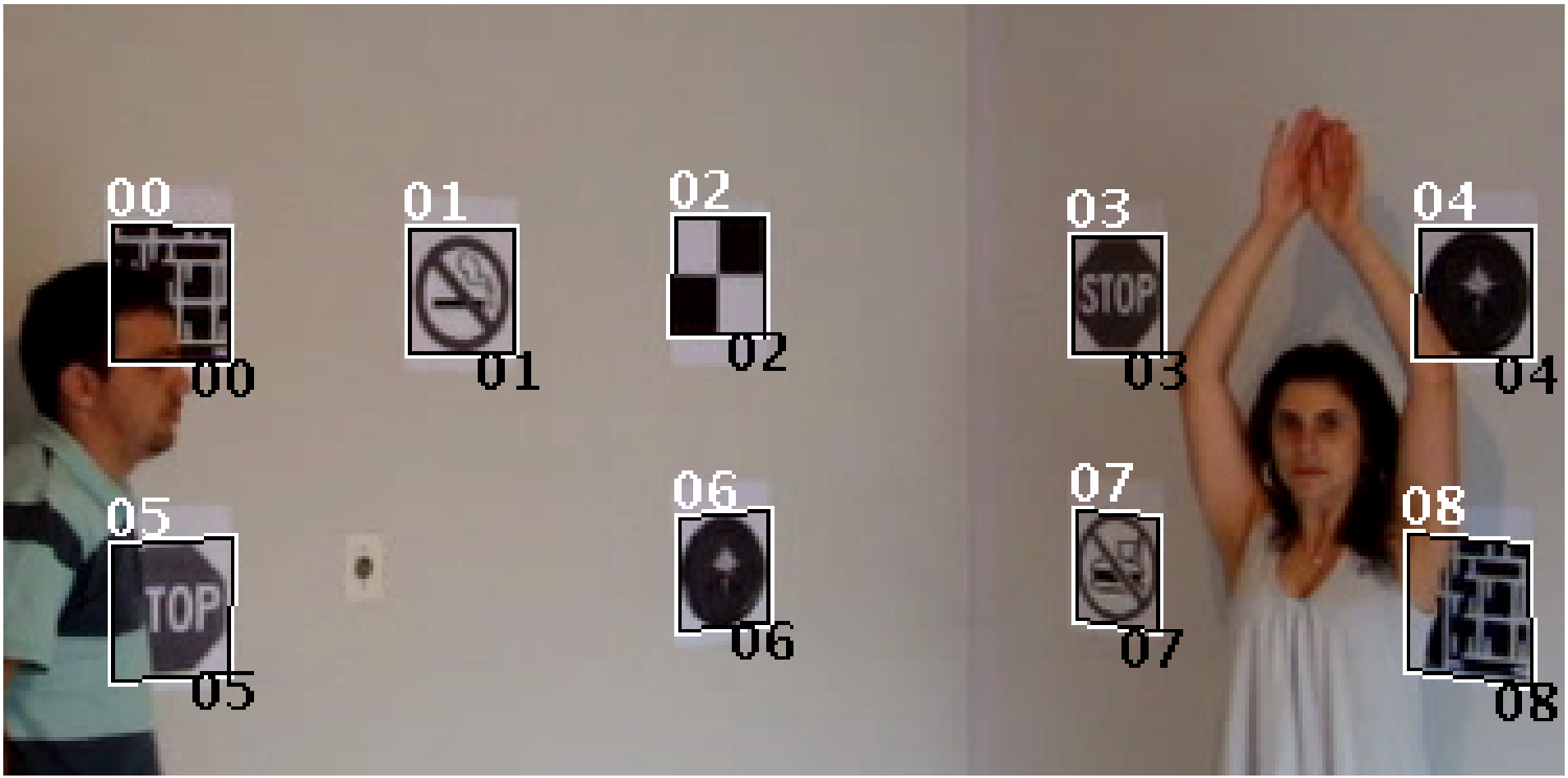
- → Feature object's geometry;
- → Canonical and masks images.



Statement of the problem

Output:

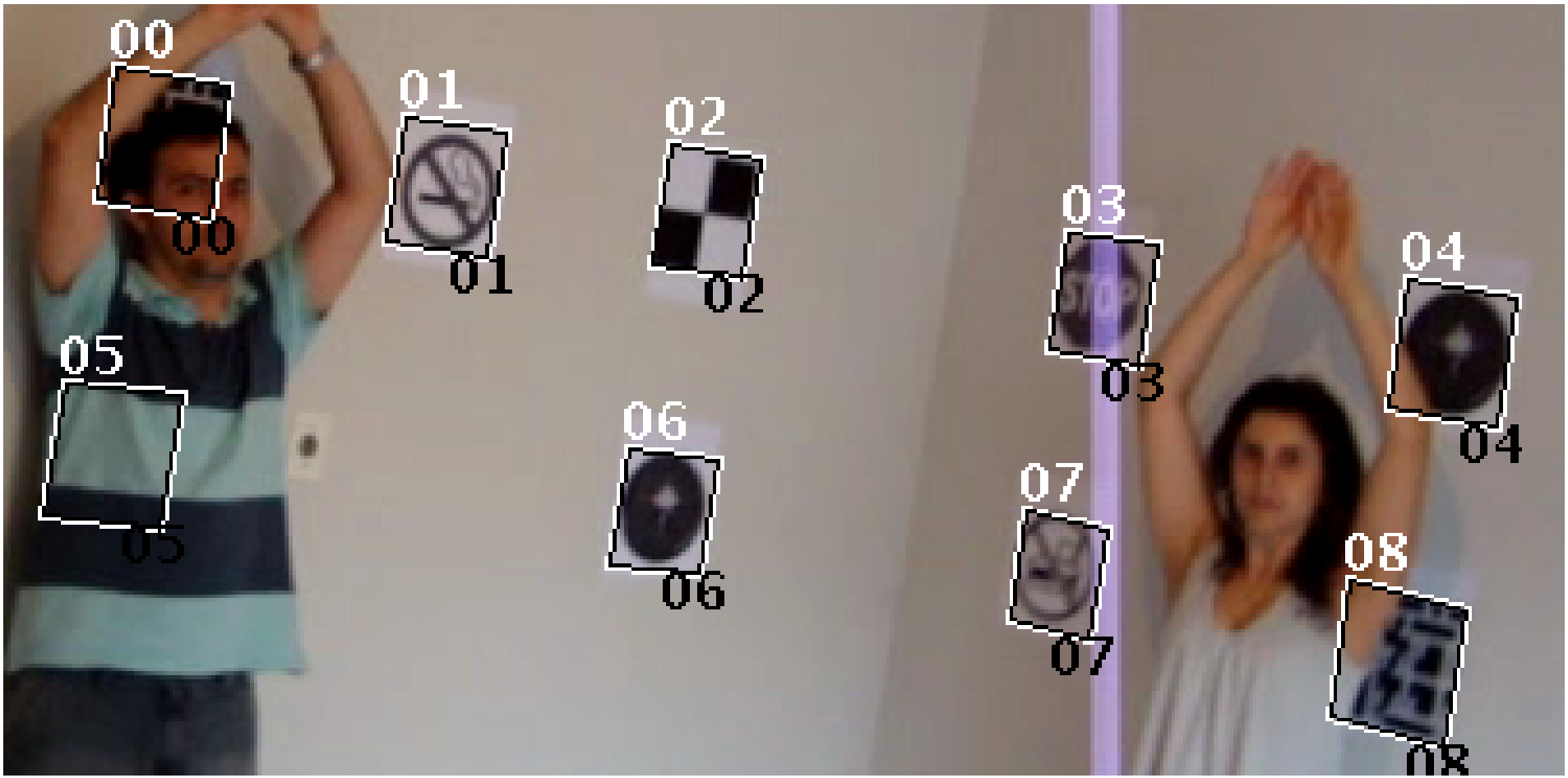
$\longrightarrow$ Feature tracking.



Statement of the problem

Output:

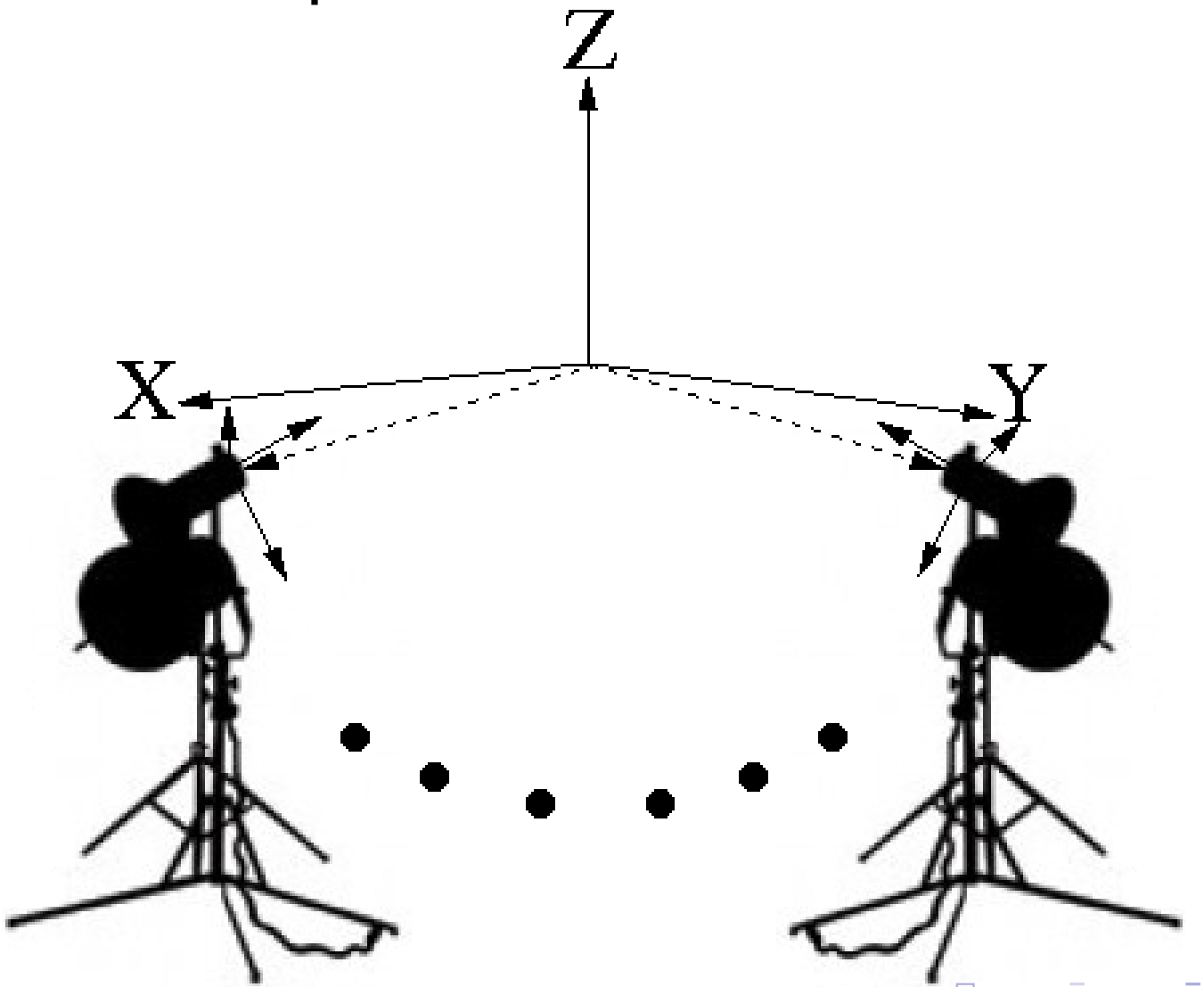
$\longrightarrow$ Feature tracking.



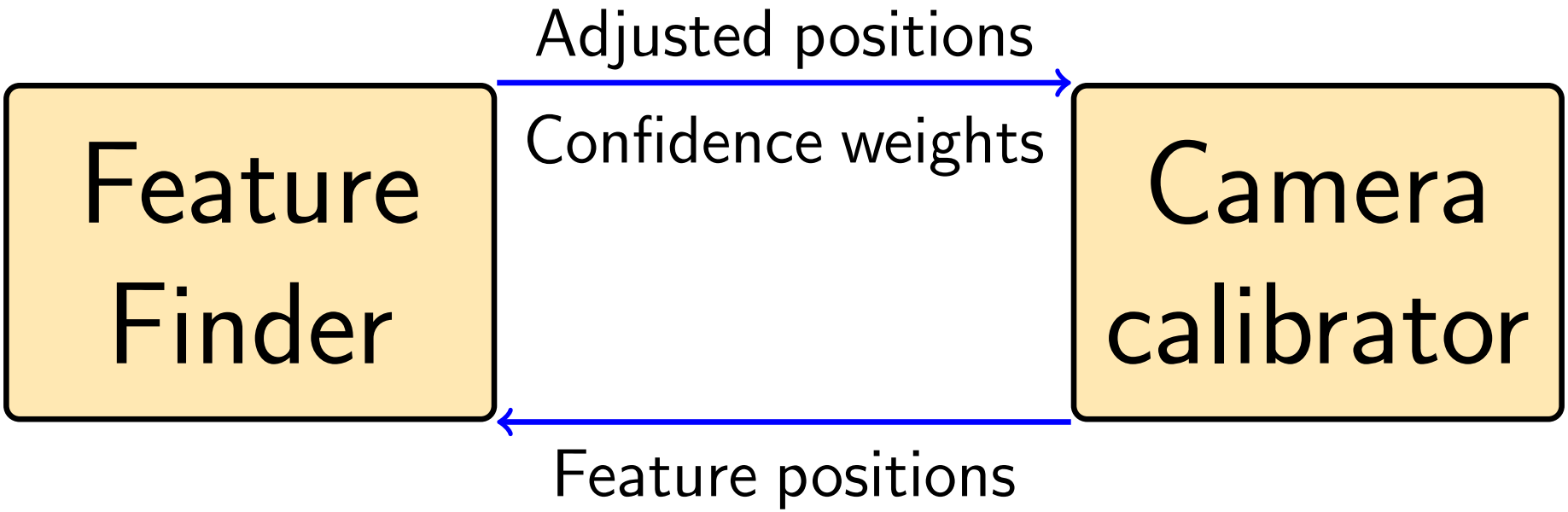
Statement of the problem

Output:

$\longrightarrow$ Camera parameters $C^{(i)}$ for each $\mathbb{V}^{(i)}$.



AFFTrack


Adjusted positions
Confidence weights
Feature Finder
Camera calibrator
Feature positions

Experiments

- 8 real videos:
  - Optura (320×240);
  - Sony (640×480);
- 9 synthetic videos:
  - POV-Ray;

Conclusions

- AFFTrack can track objects in the presence:
  - occlusions;
  - video noise;
  - tracking failures;
- AFFTrack can handle videos with:
  - variable zoom;
  - distortion zoom;
- AFFTrack is free from long-term drift;

General conclusions and perspectives

- We considered 3 computer vision problems;
- We developed 8 new algorithms:
  - T-HOG
  - SnooperText
  - Track1
  - Track2
  - Track3 - SnooperTrack
  - Track4
  - Particle Filter with T-HOG
  - AFFTrack

General conclusions and perspectives

- SnooperText/T-HOG was used in the iTowns project;
- We assembled 2 new benchmarks (publicly available);
- We plan:
  - Compare T-HOG with other descriptors (LBP, etc);
  - Improve the SnooperText algorithm;
  - Handling of occlusions in text tracking.

Publications:

[Minetto et al. 2011] **R. Minetto, N. Thome, M. Cord, J. Stolfi, F. Precioso, J. Guyomard and N. J. Leite.**

Text Detection and Recognition in Urban Scenes.

IEEE/ISPRS Workshop on Computer Vision for Remote Sensing of the Environment CVRS-ICCV .

[Minetto et al. 2011] **R. Minetto, N. Thome, M. Cord, N.J. Leite and J. Stolfi.**

SnooperTrack: Text Detection and Tracking for Outdoor Videos.

IEEE International Conference on Image Processing (ICIP).

[Minetto et al. 2010] **R. Minetto, N. Thome, M. Cord, J. Fabrizio and B. Marcotegui.**
SnooperText: A Multiresolution System for Text Detection in Complex Visual Scenes.
IEEE International Conference on Image Processing (ICIP).

[Minetto et al. 2009] **R. Minetto, N.J. Leite and J. Stolfi.**
AFFTrack: Robust Tracking of Features in Variable-Zoom Videos.
IEEE International Conference on Image Processing (ICIP).

[Minetto et al. 2008] **R. Minetto, N.J. Leite and J. Stolfi.**
Integrating Tsai's Camera Calibration Algorithm with KLT Feature Tracking.
Proceedings of IV Workshop de Visão Computacional.

Thank you for your attention!

QUESTIONS ?